

修士論文

CNN を用いた言語判定における
学習過程の分析

令和元年度修了

三重大学大学院 工学研究科 情報工学専攻
ヒューマンインターフェース研究室

富岡 永伍

はじめに

社会の国際化に伴い、カメラベース OCR を用いた自動翻訳ソフトウェアの需要が高まっている。撮影した画像から多言語の文字を同時に直接認識するカメラベース OCR の実現は困難であるため、カメラベース OCR には言語判定と呼ばれる処理が組み込まれている。言語判定とは、文字を認識する前に画像中の文字の言語を判定し、その言語専用のカメラ OCR 機能に切り替える処理のことを指し、近年は深層学習を用いた手法が主流となっている。

深層学習の登場はコンピュータビジョン分野に革命をもたらし、中でも畳み込みニューラルネットワーク (CNN) は言語判定の分野のみならず、画像解析において高い性能を発揮するため、日々応用範囲が拡大している。しかし、CNN はその不透明性から、ブラックボックス化された分類器として使われている。そこで、CNN をより良く理解するための取り組みとして、分類結果の判断根拠を可視化する手法や異なるネットワークモデルを比較するための統計的手法が数多く提案されている。これらの手法は、ユーザの CNN に対する信頼性の向上や、性能向上のために学習条件を変更する際の手助けとなっている。

本論文では、各分類クラスにおける CNN の学習過程に着目し、Loss や Accuracy の関数の設計に頑健な、異なるネットワーク同士でもクラスごとの学習過程の比較が可能である手法を提案する。現在は、ニューラルネットワークを構築する際、モデルの学習の進捗状況を知り、異なるネットワーク同士の性能を比較するために、一般的には損失関数を用いられる。しかし、損失関数から算出される値は過学習がなければ単調減少する性質を持ち、異なるネットワークモデル同士でも比較的似通った推移となることから、学習状況の比較が困難である。そこで、提案手法により損失関数を用いることなく、言語判定における CNN の学習過程の比較、ハイパーパラメータに対する依存性の理解を可能とすることを目指す。

本研究では、CNN を用いた分類で入力画像の勾配値をピクセル単位で算出する技術である Grad-CAM を応用したものを学習過程の分析用数値として扱い、この指標を“Reaction 値”と定義することで 2 種類の実験を行った。

まず一つ目の実験として、ハイパーパラメータの違いによる学習過程の分析を行った。

分析のために学習エポックに対する Reaction 値の推移グラフ, 学習画像, 評価画像それぞれの分類成功率の推移グラフや, クラスごとの損失グラフを用いることで, CNN モデルごとに各クラスの学習過程を比較した. 実験の結果, サンプルを安定して学習できるかどうかは, 畳み込み層の数や層のフィルター数といった, ハイパーパラメータに依存することが Reaction 値により評価できることが示された. また, これらのパラメータが学習サンプル数に対して十分である場合, 各クラスにおいて学習が特に活性化したエポックを明らかにすることができた. 一方で, パラメータが不十分である場合, 学習過程が不安定になるクラスが発生し, 結果としてモデルの性能にも影響をすることも確認できた.

次に, 二つ目の実験として, 交差エントロピー関数から算出される Loss 値との間で, 学習収束判定に関する性能比較を行った. Reaction 値と Loss 値それぞれにおいて, エポック間での値の変動が閾値を下回った時の各エポックを用いた場合では, Reaction 値の変動による抽出が Loss 値を用いた抽出より, 高い分類成功率をもつエポックを得ることができた. また Reaction 値, Loss 値において, 最小値をもつエポックをそれぞれ抽出したところ, 使用した 6 種類の CNN モデルすべてにおいて, Loss 値を用いた場合よりも分類成功率の高いパラメータを抽出することに成功し, Reaction 値の優位性を確認することができた. これらの結果により, Reaction 値は損失関数を用いた学習過程の分析よりも, 情報量の多い学習過程の分析が可能であり, 本手法の有効性を示すことができた.

本論文では, 1 章では研究背景と目的, 2 章では本論文に関する技術, 3 章では分析のための提案手法, 4, 5 章で分析実験と考察, 最後に 6 章で全体のまとめと今後の課題について述べる.

目次

はじめに	i
第 1 章 序論	1
1.1 研究背景	1
1.1.1 カメラベース OCR のための言語判定処理	1
1.1.2 深層学習の課題	2
1.1.3 深層学習に対する新たな取り組み	2
1.2 関連研究	2
1.3 本研究の取り組み	4
1.3.1 研究目的	4
1.3.2 研究内容	5
第 2 章 関連技術	6
2.1 Neural Networks: ニューラルネットワーク	6
2.1.1 ニューロン (神経細胞)	6
2.1.2 ニューロンモデル	6
2.1.3 Feedforward Neural Networks: 順伝播型ニューラルネットワーク	8
2.1.4 損失関数	9
2.1.5 誤差逆伝播法	9
2.2 Convolutional Neural Networks: CNN	11
第 3 章 分析手法	14
3.1 Grad-CAM の概要	14
3.2 本研究による Grad-CAM の応用 (Reaction 値)	16
3.3 Reaction 値の定義	16
第 4 章 分析実験	18
4.1 データセット	18

4.2	CNN の学習	20
4.3	実験概要	22
4.3.1	ハイパーパラメータ変更による学習過程の比較	22
4.3.2	Reaction 値, Loss 値を用いた最適エポックの抽出	22
第 5 章	結果と考察	24
5.1	ハイパーパラメータの違いによる学習過程分析実験	24
5.1.1	畳み込み層の数の違い	24
5.1.2	畳み込み層のフィルター数の違い	28
5.2	最適エポック抽出実験	33
5.2.1	学習収束判定の性能比較	33
5.2.2	最小値による分析成功率の比較	36
第 6 章	結言	37
6.1	まとめ	37
6.2	今後の課題	38
付録 A	各モデルにおける言語ごとの分析結果	39
A.1	畳み込み層の数の変更による分析結果	39
A.2	畳み込み層のフィルター枚数の変更による分析結果	46
付録 B	検証データを用いた Reaction 値の性能評価	51
B.1	実験概要	51
B.2	結果と考察	51
付録 C	Grad-CAM を用いた CNN の判断根拠の可視化	54
C.1	実験概要	54
C.2	結果と考察	54
付録 D	研究で用いたデータの参照場所	56
付録 E	発表資料	57
	謝辞	60

第 1 章

序論

1.1 研究背景

1.1.1 カメラベース OCR のための言語判定処理

近年、スマートフォンやタブレット PC などの普及に伴い、誰もが手軽に写真を撮影できるようになった。気軽に撮影できる点から、スナップ写真を撮るだけでなく、メモや掲示物の情報をカメラで撮影して記録する用途にも使われつつある。また、カメラベース OCR (Optical Character Recognition) と呼ばれる、カメラ画像の文字認識を行う技術の高精度化や高速化によって、情景内文字に対する自動翻訳ソフトウェアが徐々に実用化されている。更に、社会のグローバル化により世界中を行き来する人が増加していることから、カメラベース OCR には多言語対応が必要不可欠となっている。多言語対応を実現する場合、文字を認識する前に画像中の文字の言語を判定し、その言語専用のカメラ OCR 機能に切り替えるほうが、高い認識精度を期待できる [1]。言語判定機能は数十言語に対応する翻訳ソフトウェアにおいては、通常搭載されている。文書画像や情景文字画像のような文字に対する画像認識問題は、データセットに含まれる同一クラスのサンプル同士の見かけ上の差異が一般の画像認識よりも小さく、学習が容易であるため、比較的規模が小さいニューラルネットワークでも取り扱うことが可能である。そのため近年は、CNN などのニューラルネットワークを用いた手法 [2, 3, 4] が主流となっておりモバイル機器などリソースに制限のあるデバイスへの搭載が期待されている。

1.1.2 深層学習の課題

深層学習の登場はコンピュータビジョンに革命をもたらし、中でも畳み込みニューラルネットワーク (Convolutional Neural Networks : CNN [5]) は言語判定の分野に留まらず、画像解析において高い精度が得られるため [6]、日々応用範囲が拡大している。CNN は様々な画像認識問題において容易に使用することが可能であるが、精度の向上に伴い、より層が深く、より複雑な構造になりつつある。よって CNN を始めとするニューラルネットワークには、入力データから出力への変換過程が不透明であることや、誤認識に対する原因考察が困難なことなど、精度と透明性がトレードオフの関係であるという問題を抱えている [7, 8].

1.1.3 深層学習に対する新たな取り組み

ブラックボックス化された分類器として扱われる CNN に対し、処理やパラメータを解明する重要性が世界各国の研究者によって言及されつつある。深層学習にはモデルの性能に影響を及ぼす、調整可能なパラメータが数多く存在する。これらはハイパーパラメータと総称され、モデルの複雑化に伴いその数も増加するので、手作業や簡単な手法では細かい調整が手に負えない状況となる。そこでハイパーパラメータ調整のための最適化手法や、パラメータ変更によるモデルの性能の有効性を、可視化等を用いて検証した研究が多数報告されている [9, 10, 11]。また、図 1.1 に示す Saliency maps [12] や Grad-CAM [13], SmoothGrad [14] のようなフレームワークによって、CNN を用いた分類における判断根拠を可視化出来るようになった。可視化することによって、ニューラルネットワークの性能改善の手がかりが得られる見込みがあることや、ユーザからの信頼を確立するなど様々な利点がある。これらの手法は、物体認識や画像キャプション生成、VQA (Visual Question Answering) など多くの画像認識分野で使用され、CNN をより良く理解し、性能を効果的に向上させるための橋渡しの役目を果たしている [15].

1.2 関連研究

ニューラルネットワークの処理やパラメータを解明する研究に伴い、異なるニューラルネットワークの構造や学習状態を分析し、それらを比較するための多くのアプローチがなされている。

Raghu らは、異なるネットワーク構造を持つニューラルネットワーク同士を比較できる “Singular Vector Canonical Correlation Analysis (SVCCA)” を提案した [17]。入力には 2 つのネットワークの層が与えられ、各部分間の特異値を分解し、部分空間の 99% の分散

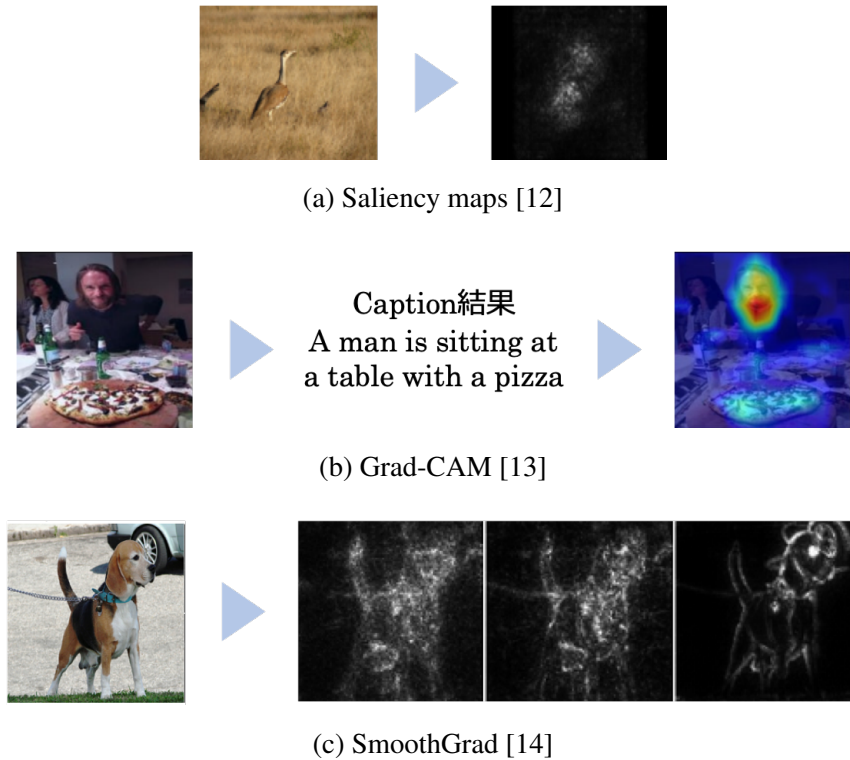


図 1.1: CNN の仕組みを可視化するフレームワーク

が表現可能な方向を取得する．次に，互いのネットワークの正準相関の類似性を計算した後 [18]，線形変換した部分空間の相関係数を出力する．SVCCA の柔軟性を用いることで，様々なランダム初期化，ネットワーク構造，学習のステップ，特定のクラスと層の特徴が比較されている．

Li らは，“Filter-Wise Normalization”を提案することで，ニューラルネットワークの損失関数の構造を可視化した [19]．互換性のある次元を持つランダムなガウス方向ベクトルを生成し，そのベクトルからフィルタごとに正規化された方向を取得することで，図 1.2 に示すような損失関数の可視化を実現した．これらの可視化を活用し，ネットワーク構造，オプティマイザ，バッチサイズ等の違いによる損失関数の影響が調査されている．

また，Shenk らは，ニューラルネットワークの層単位の飽和状態を学習中に算出できる手法を提案した [22]．主成分分析によって近似された固有次元を特徴空間の次元と比較することによって，ネットワークの各層の飽和度を推定する．Raghu らが提案した SVCCA とは対照的に，シンプルで高速な計算手法であるため，ネットワークに対してどの程度適切なパラメータであるのかを，学習中に比較できる点がこの手法の利点である．

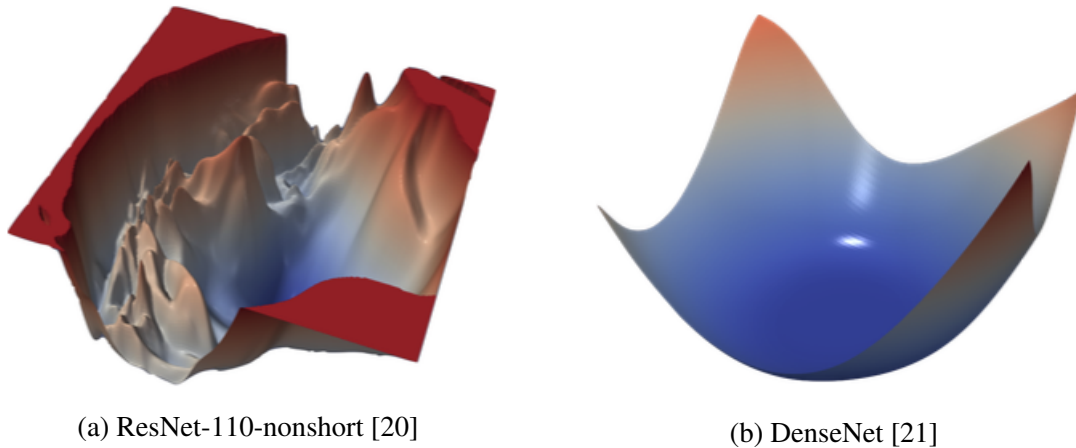
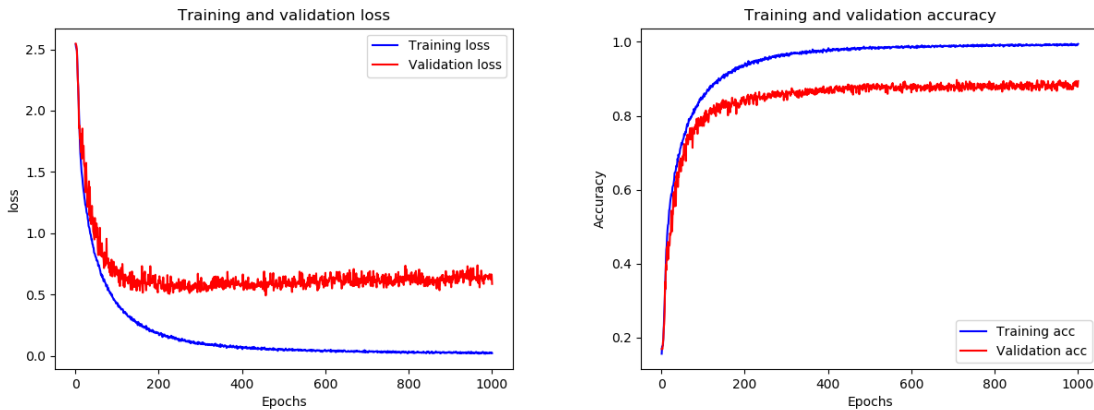


図 1.2: 損失関数の曲率の可視化

1.3 本研究の取り組み

1.3.1 研究目的

CNN の内部状態を解析することで、より効率的に学習用のパラメータを定めることができる。我々がネットワークモデルを構築する際、構築モデルの学習の進捗状況を知り、ネットワークの性能をより良くするには、一般的には損失 (Loss) 関数を用いる [16]。ネットワークの学習には、通常訓練用データに加え検証用データと呼ばれるデータを用いて、この2種類のデータの損失、精度を同時に確認することで、評価用データに対して汎化性の高いモデルを構築する。図 1.3 にネットワークの学習の進捗状況を確認する際に用いられる、損失と精度の推移グラフを示す。図 1.3 (a) は、訓練用データと検証用データそれぞれにおいて、学習を進めた時の損失の推移を表したグラフ、図 1.3 (b) は訓練用データと検証用データそれぞれの精度を表したグラフであり、学習を繰り返すことで損失の値が下がり、その分精度は上昇する。これらの値を確認することで、モデルの学習が十分であるか、過学習は起きていないかなど、作成する CNN の学習の進捗状況を理解することができるため、損失関数はより良いネットワークモデルを構築するための必要不可欠な分析指標となっている。しかし、損失関数から算出される値は過学習がなければ単調減少する性質をもつため、異なる構造を持つネットワーク同士でも比較的似通った値の推移となる。よって、モデル間の学習状況の比較が容易ではなく、またネットワークが特に活性化している学習エポックを特定することも困難である。そこで、本研究では CNN の学習過程の分析を、損失関数を用いることなく行い、異なるモデル間の学習過程の明確な比較やハイパーパラメータに対する依存性を理解することを目的とする。



(a) 損失 (Loss) の推移グラフ

(b) 精度 (Accuracy) の推移グラフ

図 1.3: ネットワークモデルの学習状況を確認するための 2 つのグラフ

1.3.2 研究内容

本研究では損失関数に代わる，CNN の学習進捗状況进行分析するための手法を提案する．CNN を用いた分類で入力画像に対する勾配値をピクセル単位で算出する技術である Grad-CAM を応用することで，CNN の学習過程进行分析する．また，前節で述べた言語判定問題は，クラス数が他の分類問題と比べて比較的少ないので，提案した手法が学習過程の分析に適しているかを検証するには，程よい分類問題であると考えられる．よって，本研究では言語判定問題における学習過程に着目し，CNN の分析を行う．分析する CNN モデルは，畳み込み層の数や層ごとのフィルター数など，任意のハイパーパラメータのみを変更し，それ以外のハイパーパラメータは統一したモデルを数種類作成する．言語判定の分野で広く用いられているデータセット SIW-13 において，CNN の学習性能が，ハイパーパラメータにどの程度依存しているかを調査すると共に，異なる CNN モデルの学習過程との比較も行う．また，交差エントロピー関数から算出される Loss 値を用いる場合と，学習の収束判定に関する性能比較も行うことで，本手法の有効性を示す．

第 2 章

関連技術

2.1 Neural Networks: ニューラルネットワーク

ニューラルネットワークとはニューロンモデル (ユニット, ノード) を多層的に結合したモデルである。ニューラルネットワークは入力層, 出力層, 隠れ層から構成され, 脳のニューロン (神経細胞) をモデル化したものである。

2.1.1 ニューロン (神経細胞)

ニューロンは複数の受信機 (樹状突起: dendrite) と一つの送信機 (軸索: axon) で構成され, 軸索上を伝わる電気パルスによってその他のニューロンへと情報が伝達される。軸索は, シナプスと呼ばれるインタフェースを介して, パルスの到来をニューロンに伝達する。

ニューロンは電子パルスを受け取ることで, 細胞内の電気レベル (膜電位) が上下する。この変動は, 入力を受け取るシナプスの状態 (シナプス伝達強度) に依存する。そして, 膜電位の値がある一定の値を超えると, その電子パルスは発信され, 軸索を通して他のニューロンに伝達される。

2.1.2 ニューロンモデル

ニューロンモデルはニューロンを単純な数理化モデルで表現したものであり, 以下の式 (2.1), (2.2) のような入出力関係になる。ここで, x_1, x_2, \dots, x_n をニューロンへの入力, w_1, w_2, \dots, w_n をシナプス伝達強度, b をバイアス, z をニューロンの出力とする。出力 z は次のニューロンへの入力となる。

$$y = \sum_{i=1}^n w_i x_i + b \quad (2.1)$$

$$z = f(y) \quad (2.2)$$

この時、 $f()$ は非線形関数であり、活性化関数と呼ばれる。活性化関数の役割は、ニューロンの応答に非線形性を与えることである。以下に、4種類の活性化関数を示す。

- ReLU 関数

ReLU (Rectified Linear Unit) 関数は以下の式 (2.3) で定義される。

$$f(x) = \begin{cases} x & (x \geq 0) \\ 0 & (x < 0) \end{cases} \quad (2.3)$$

入力値が 0 以上の場合は入力値がそのまま出力値となり、0 以下の場合は 0 となる。ReLU 関数は CNN の畳み込みフィルターや、全結合層の後に置かれ、抽出された特徴をより強調する働きがある。

- シグモイド関数

シグモイド関数は以下の式 (2.4) で定義される。

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (2.4)$$

入力値 x が小さいほど出力値は 0 に近づき、大きいほど出力値は 1 に近づく。シグモイド関数は 2 クラスの識別問題の場合によく用いられる活性化関数である。

- softmax 関数

softmax 関数は以下の式 (2.5) で定義される。

$$f(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}} \quad (2.5)$$

softmax 関数は i 個存在する入力値 x をそれぞれ確率分布に正規化することで、入力値の総和を 1 となるように算出する活性化関数である。そのため、CNN では多クラス問題の識別時に出力層として、softmax 関数がよく用いられる。

- 恒等関数

恒等関数は以下の式 (2.6) で定義される。

$$f(x) = x \quad (2.6)$$

恒等関数は入力した値を変換することなくそのまま出力するため、回帰問題などで用いられる場合がある。

2.1.3 Feedforward Neural Networks: 順伝播型ニューラルネットワーク

順伝播型ニューラルネットワークは単純パーセプトロンを並べたものを一つの層とし、隣接した層を結合したものであり、多層パーセプトロン (multilayer perceptrons: MLPs) とも呼ばれる。多層パーセプトロンは、入力層、中間層、出力層の3種類のニューロンモデルから構成される。図 2.1 のように多層パーセプトロンは入力データに線形変換と活性化関数による非線形変換を繰り返し行うことで、任意の関数を近似することができる。中間層の数が多くなるほどネットワークの表現力は大きくなるのが分かっているが、入力に高次元の特徴ベクトルを入力する場合、ネットワークのパラメータ数が急激に増大し、学習が困難となる。

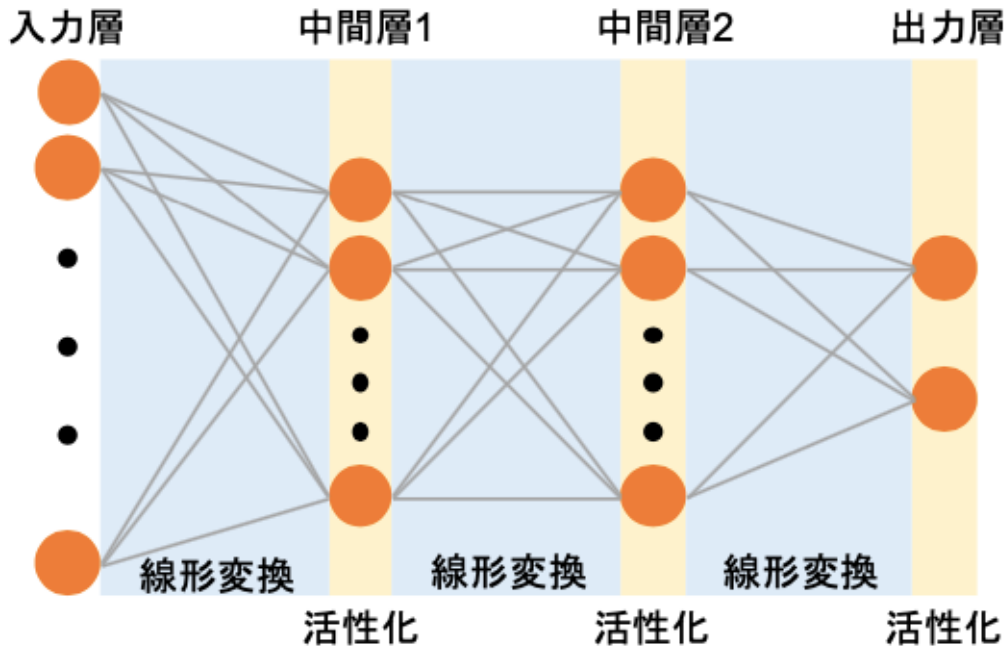


図 2.1: 中間層 2 つの多層パーセプトロンの例

2.1.4 損失関数

損失関数とは、ニューラルネットワークの性能の悪さを表す指標であり、訓練データと正解データの間にはどれほど誤差があるかを計算することができる。ニューラルネットワークの学習における損失関数は主に 2 乗和誤差 (2.7) や交差エントロピー誤差 (2.8) が用いられる。

$$E = \frac{1}{2} \sum_k (y_k - t_k)^2 \quad (2.7)$$

$$E = - \sum_k t_k \log y_k \quad (2.8)$$

ここで、 E は出力される損失 (Loss) 値、 y_k 、 t_k はそれぞれ、ネットワークが出力したクラス k の事後確率の値、出力された値に対応する正解値である。

2.1.5 誤差逆伝播法

ニューラルネットワークは損失関数で得られた値が小さくなるように重みパラメータ w を更新することで、モデルの最適化を行う。

誤差逆伝播法 (Backpropagation) [23] は、多層型ニューラルネットワークを学習する際に用いられるアルゴリズムであり、ネットワーク上の変更可能な重みについて、学習時に発生する誤差の傾斜を計算することができる。入力層から順伝播を計算し、出力された全ての出力ノードから入力ノードの方向へと誤差を逆伝播することで、入力に近いノードの重みを更新する。以下に誤差逆伝播法による重み更新の流れを示す。ここで、第 n 層のニューロンの数を L_n 個と表すことで、第 p 番目の入力信号を x_{pL_1} 、第 p 番目の教師信号を t_{pL_n} と表し、第 j 番目のニューロンの出力を y_j^n と表す。

1. 初期値として、全ての重みを乱数によって -0.1 から 0.1 程度の範囲の小さな値に設定し、学習率 η を 0 から 1 の間で設定する。
2. 入力信号 $x_{pi}(1 \leq i \leq L_1)$ をネットワークに入力する。
3. 入力層から出力層に向けて、各ニューロンの出力を計算する。
4. 教師信号 t_{pj} の誤差と出力層の出力 y_j^N から以下の (2.9) 式によって、 δ_j^N を計算する。

$$\delta_j^N = -(t_{pj} - y_j^N) y_j^N (1 - y_j^N) \quad (2.9)$$

5. δ_j^N を用いて、中間層の誤差信号 δ_j^n を以下の (2.10) 式によって計算する。ただし、 $(n < N)$ とする。

$$\delta_j^n = \left\{ \sum_{k=1}^{L_{n+1}} \delta_j^N w_{k,j}^{N;n} \right\} y_j^n (1 - y_j^n) \quad (2.10)$$

6. δ_j^n を用いて、以下の (2.11) 式によって重みを更新する.

$$\Delta w_{j,i}^{n;n-1} = -\eta \delta_j^n y_j^{n-1} \quad (2.11)$$

誤差逆伝播法は、一般的に確率的最急降下法を用いることで、重みとバイアスを修正し、高速に損失関数を最小化できるため、高い汎用性を有している.

2.2 Convolutional Neural Networks: CNN

畳み込みニューラルネットワーク (Convolutional Neural Networks : CNN) は少ないパラメータで構成される畳み込み層と多層パーセプトロンとを組み合わせたニューラルネットワークである。この技術は 1990 年代初期から文字認識分野で使用されていた [24]。現在のブームは物体認識技術を競う “ImageNet large-scale visual recognition challenge 2012” において CNN を用いた手法 [5] が高い認識精度を誇ったことがきっかけで起こった。畳み込み部と多層パーセプトロン部が組み合わされたネットワークモデルであっても、全てのパラメータに対して誤差関数との微分が可能であれば、誤差逆伝播法によって end-to-end で学習できることが CNN の大きな利点である。そのため、近年では、単純なネットワーク構成の設計の他に、誤差関数の設計を工夫することで、画像認識の他にセマンティックセグメンテーション (U-Net [25], Fully Convolutional Networks [26]) や画像生成 (Generative Adversarial Network [27]) など新しい分野にも応用されている。

畳み込みニューラルネットワークの構成

CNN の入力通常、3 階のテンソル型である。この入力に対して、畳み込み層での特徴マップ生成、プーリングを交互に行い、前層からの入力情報を次の層へ伝播させる。特徴マップが含む入力から抽出された形状特徴をプーリング処理によって縮小しつつ、上位の層への伝播が可能なモデルである。その後、全結合層、出力層によってモデルが推定した事後確率ベクトルを出力する。

入力層

サイズ $W(\text{Width}) \times H(\text{Height}) \times D(\text{Depth})$ の画像を入力する。

畳み込み層

畳み込み層では、画像の一部とフィルタの要素積の和を、画像をスライドさせながら画像の全領域で求める。畳み込み処理の概略図を図 2.2 に示す。ここでの畳み込み演算において、フィルターを適用する位置の間隔 (ストライド) のサイズは 2 とする。

また、 l 層目の入力 x^l に対してストライドが 1 の時の畳み込み処理は式 (2.12) で定義できる。ここで、バイアスは加えないものとする。

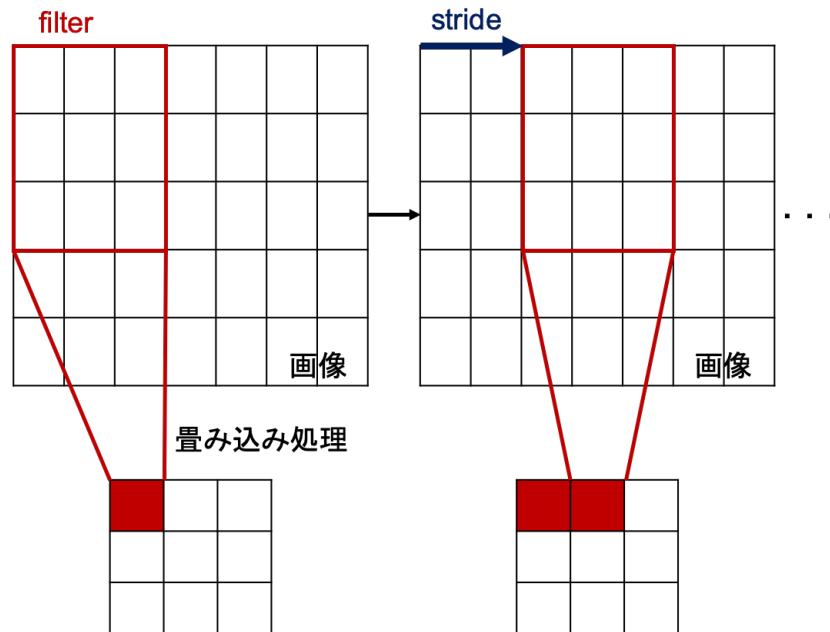


図 2.2: 畳み込み処理の概略図

$$y_{i^{l+1}, j^{l+1}} = \sum_{p=1}^H \sum_{q=1}^W f_{p,q} \times x_{i^l+p, j^l+q} \quad (2.12)$$

H , W はそれぞれフィルターの縦, 横のサイズ ($0 \leq i < H$, $0 \leq j < W$), $y_{i^{l+1}, j^{l+1}}$ は l 層目で出力された特徴マップ y の i 行 j 列目の要素, $f_{p,q}$ はフィルターの p 行 q 列目の要素を表す.

畳み込み処理と活性化を繰り返すことによって, 入力画像を認識に有効となる多チャンネル特徴マップへと変換することができる.

最大プーリング層 (Max pooling layer)

最大プーリング層は, 任意の大きさの領域内の最大値をとり, 画像の圧縮を行う層である. 畳み込み層で得られた各特徴マップに対して, 画像中の対象物の位置不変性を確保するために, 前節で先述した ReLU 関数 (2.3) を通して, Pooling 層に入力する処理が一般的である. 図 2.3 に特徴マップサイズが 4×4 , 局所領域サイズが 2×2 の場合のプーリング処理の例を示す.

また, l 層目のフィルターサイズ $W^l \times H^l$ の入力 x^l に対してのプーリング処理は式 (2.13) で定義できる.

$$y_{i^{l+1}, j^{l+1}} = \max_{0 \leq i < H, 0 \leq j < W} x_{i^{l+1} \times H + i, j^{l+1} \times W + j}^l \quad (2.13)$$

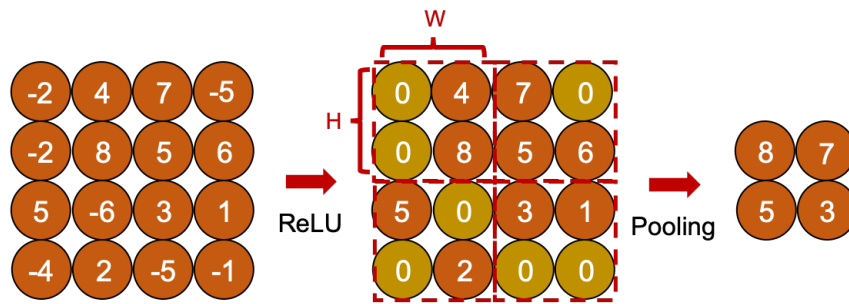


図 2.3: ReLU 関数とプーリング処理

ここで、 $0 \leq i < H$, $0 \leq j < W$, $y_{i+1,j+1}$ は $l+1$ 層目の出力 y の i 行 j 列目の要素を表す。畳み込み処理の間にプーリングを挟むことで特徴マップのサイズが圧縮され、その後の全結合層 (多層パーセプトロン部) の計算コストとパラメータ量を大きく減少させる効果がある。

全結合層 (Fully connected layer と出力層 (Output layer))

全結合層 (多層パーセプトロン) は、畳み込み層とプーリング層を通して特徴部分を取り出された画像データを一つのノードに結合し、活性化関数によって変換された値を出力する。入力層と出力層の間に全結合層を入れることによって、全結合層のノード数の分だけ特徴空間の分割数が増し、各領域を特徴づける変数が増えるため、より正確な分類が期待できる。

出力層では、全結合層からの出力を元に、softmax 関数を用いて確率に変換することで分類を行う。

第 3 章

分析手法

ネットワークのノード間の勾配を算出することで、ニューラルネットワークの学習過程を分析することが可能である。そこで本研究では、誤差逆伝播法における勾配を、クラスごとに算出する技術である Grad-CAM (Gradient-weighted Class Activation Mapping) [28] を用いる。Grad-CAM で用いられる計算式を利用して、CNN の学習過程を分析する。

3.1 Grad-CAM の概要

Grad-CAM は CAM (Class Activation Mapping) [29] の拡張手法であり、CNN を用いる識別において活性化したニューロンに対応する箇所を可視化する技術である。Grad-CAM はクラスごとの確率スコアへの影響が大きい領域を微分係数の平均化によって特定するので、分類時に画像中で勾配の高い箇所を特定することができる。ここでの微分係数は変化率を表すという意味で勾配とも呼ばれることから Grad-CAM と呼ばれるようになった。Grad-CAM の利点として、ネットワークの制限や変更が必要ないことがあげられる。CAM とは異なり、ネットワークの構造を変える必要がないので、既存の複雑な構造のネットワークや自作したネットワークでも分析が可能である。また、対象クラス以外の勾配を 0 にすることで必要なクラスのみ勾配を求めることができる点も Grad-CAM の利点の 1 つである。以下の式によって、クラスごとの勾配が算出される。

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y'^c}{\partial A_{ij}^k} \quad (3.1)$$

$$L_{Grad-CAM}^c = ReLU \left(\sum_i \alpha_k^c A^k \right) \quad (3.2)$$

ここで、 c は可視化対象クラス、 k はフィルタ数、 y'^c は全結合層の最終出力であり *softmax* 関数を適用する前の値である。式 (3.1) では、クラス c の確率スコア y'^c を k 番目の特徴マップの (i, j) ピクセルにおける強度 A_{ij}^k について微分して勾配 $\partial y'^c / \partial A_{ij}^k$ を計算し、そ

れらを全ピクセルについて平均することにより，クラス c の k 番目のフィルタに関する重み係数 α_k^c を算出する．この重み係数 α_k^c が大きいほど，その特徴マップ A^k がそのクラス c にとって重要であることを意味する．式 (3.2) では，式 (3.1) で算出された重み係数 α_k^c により k 個のフィルタの加重平均を算出する．Grad-CAM は正の影響を与えた部分のみを特定するため，その活性化関数 $ReLU(x) \equiv \max(0, x)$ による出力をヒートマップ出力に変換する．

図 3.1 にアラビア語の情景内画像を入力した時の Grad-CAM の出力例を示す．図 3.1 (a) を入力し，勾配を求めたいクラスとしてアラビアクラスを指定した時の画像が図 3.1 (b) である．Grad-CAM は最終層で Global Average Pooling (GAP) [30] を行うことで，指定クラスの重要度を可視化できる．出力されるヒートマップ画像では，暖色の部分ほど式 (3.2) で算出される値が大きく，畳み込み層の間で高い勾配値をもつ，すなわちネットワークの勾配が高い箇所に対応しているピクセルである．したがって，図 3.1 (a) をアラビア語として分類した時，図 3.1 (b) から，ヒートマップ画像の暖色部分にあたる，主にアラビア文字の横線の部分に対してネットワークのニューロンの重みが大きくなったということがわかる．



図 3.1: Grad-CAM の例

3.2 本研究による Grad-CAM の応用 (Reaction 値)

前節で記述した Grad-CAM による勾配値を利用することで、言語判定における CNN の学習の進行状況を分析する。Grad-CAM はクラスごとの決定境界の違いを可視化するために、入力画像の予測スコアに対する特徴マップへの重み係数を各ピクセルで算出する。そのため、Grad-CAM は本来未学習の評価画像に対して適用され、分類の決め手となったピクセルを特定することに用いられる。さらに入力画像に対する CNN の特徴マップのクラスごとの重みを明らかにできるため、学習画像に Grad-CAM を適用すれば、各ピクセルで得られた勾配値を平均化することで、画像単位のニューロンの重みを学習の進行状況として置き換えて算出することができる。

そこで本研究では、一般的に評価画像に対して用いられる Grad-CAM を CNN を構成する際に用いる学習画像に適用する。Grad-CAM は入力画像の各ピクセルで重み係数 (勾配) を算出するので、1 枚の画像につき、画像の縦横サイズのみだけ勾配値を持つ。画像中の正の勾配をもつピクセルのみの値を平均したものを本研究では“Reaction”値と定義し、この値を各クラスの全学習画像で算出することで学習の進行状況を分析する。

3.3 Reaction 値の定義

Reaction 値は前節の Grad-CAM の式に倣って以下の式 (3.3) で定義する。

$$Reaction = \frac{1}{pN} \sum_i \sum_j L_{ij}^c \quad (3.3)$$

ここで、 c は可視化対象クラス、 i, j はそれぞれ CNN の特徴マップの縦、横サイズ、 pN は特徴マップ中で正の勾配を持つピクセルの数である。入力画像の正の勾配値を持つピクセルのみで平均をとることで、特徴マップのサイズに依存することなく、その画像におけるニューロン活性化の度合いを統計的に求められる。学習が進んでいない段階ではニューロンが活性化していないので Reaction 値は低い値となる。学習が進めばニューロンが次第に活性化するため Reaction 値は増加し、ニューロンの活性化後は学習の変化が収束するので Reaction 値は減少していくと予想できる。そのため、CNN モデルに対する任意のクラスにおいて、学習が活性化しているエポックが明らかになり、他クラス、他モデルとの比較も容易になり得る。

図 3.2 にアラビア語を含む情景画像を入力した時の Reaction 値の算出の流れを示す。入力画像に対する勾配値のヒストグラムより、ピクセルごとに勾配値はさまざまであることがわかるが、全体で平均を取ることで得られる Reaction 値を、入力画像に対する学習の活性度として扱う。Reaction 値は、ネットワークの学習進行状況を分析する数値指標と

して定義したが、Grad-CAM と同様に、評価用画像や検証用画像にも適用できるため、本研究で対象としている言語判定問題における学習進行状況の分析以外にも応用可能だと考える。

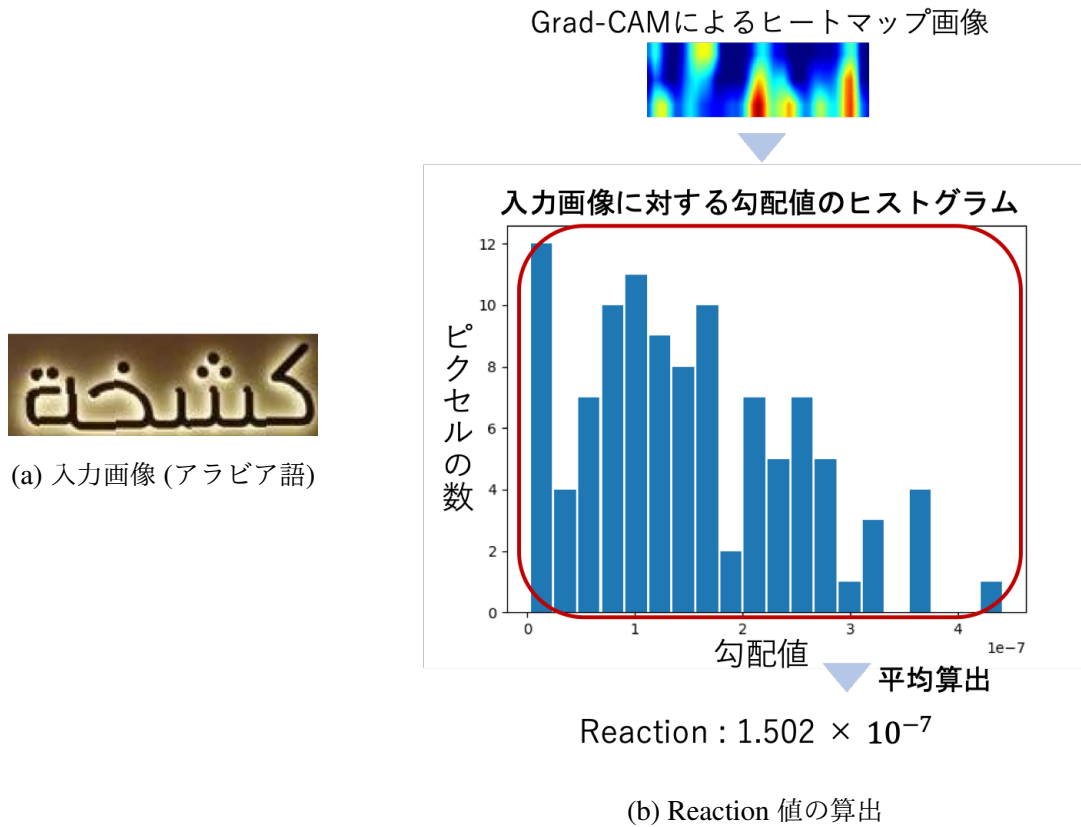


図 3.2: Reaction 値の算出の流れ

第 4 章

分析実験

4.1 データセット

本研究では，情景文字画像のデータセット，SIW-13 [31] を用いて評価実験を行う．図 4.1 に SIW-13 の画像例を示す．SIW-13 は，Google Street View [32] から文字部分を切り出した画像であり，13 言語，全 16291 枚の文字画像で構成されている．対象クラスは，アラビア語，カンボジア語，中国語，英語，ギリシャ語，ヘブライ語，日本語，カナダ語，韓国語，モンゴル語，ロシア語，タイ語，チベット語の 13 クラスである．これらの画像は全て情景画像から切り取ったものであることから，文字の角度，フォント，色，大きさが画像によって異なっている．また画像によっては，ぼやけた画像や光やカメラの向きの影響を受けた画像も存在することから，データセットがより実環境で起こりうる変化に即したものになっている．さらに，ICDAR2011 [33]，SVT [34]，IIIT 5K-Word [35] のような従来のデータセットに比べ，含まれる言語の数も 13 クラスと多いため，言語判定の研究における主要な研究用のデータセットとなっている．表 4.1 に SIW-13 の各クラスの学習画像数，評価画像数の内訳を示す．SIW-13 は予め学習画像と評価画像が分割されており，学習画像は全 9791 枚，評価画像は各言語 500 枚の全 6500 枚である．本実験では，ネットワークの汎化性能を測るため，各クラスの学習画像からそれぞれ 1 割を検証用画像として扱い学習を行った．



図 4.1: SIW-13 の画像例

表 4.1: SIW-13 データセットの内訳

言語	画像枚数	学習データ枚数	評価データ枚数
アラビア語	1,002	502	500
カンボジア語	1,083	583	500
中国語	1,298	798	500
英語	1,221	721	500
ギリシャ語	1,018	518	500
ヘブライ語	1,242	742	500
日本語	1,215	715	500
カンナダ語	1,029	529	500
韓国語	1,561	1,061	500
モンゴル語	1,192	692	500
ロシア語	1,031	531	500
タイ語	2,222	1,722	500
チベット語	1,177	677	500
総画像枚数	16,291	9,791	6,500

4.2 CNN の学習

本研究では，分類用の CNN モデルとして，畳み込み層の数や層のフィルタ数が異なる学習モデルを 6 種類用意する．6 つの CNN モデルの畳み込み，プーリング，特徴マップの 1 次元変換部分の詳細を図 4.2 に示す．

Layer1 : Conv 16x3x3
Max pooling 2x2 dropout 0.5
Layer2 : Conv 32x3x3
Max pooling 2x2 dropout 0.5
Layer3 : Conv 64x3x3
Max pooling 2x2 dropout 0.5
Flatten 5888

(a) Model I

Layer1 : Conv 32x3x3
Max pooling 2x2 dropout 0.5
Layer2 : Conv 64x3x3
Max pooling 2x2 dropout 0.5
Layer3 : Conv 128x3x3
Max pooling 2x2 dropout 0.5
Flatten 11776

(b) Model II

Layer1 : Conv 16x3x3
Layer2 : Conv 16x3x3
Max pooling 2x2 dropout 0.5
Layer3 : Conv 32x3x3
Layer4 : Conv 32x3x3
Max pooling 2x2 dropout 0.5
Layer5 : Conv 64x3x3
Layer6 : Conv 64x3x3
Max pooling 2x2 dropout 0.5
Flatten 2688

(c) Model III

Layer1 : Conv 32x3x3
Layer2 : Conv 32x3x3
Max pooling 2x2 dropout 0.5
Layer3 : Conv 64x3x3
Layer4 : Conv 64x3x3
Max pooling 2x2 dropout 0.5
Layer5 : Conv 128x3x3
Layer6 : Conv 128x3x3
Max pooling 2x2 dropout 0.5
Flatten 5376

(d) Model IV

Layer1 : Conv 16x3x3
Layer2 : Conv 32x3x3
Layer3 : Conv 32x3x3
Max pooling 2x2 dropout 0.5
Layer4 : Conv 64x3x3
Layer5 : Conv 64x3x3
Max pooling 2x2 dropout 0.5
Layer6 : Conv 128x3x3
Layer7 : Conv 128x3x3
Layer8 : Conv 256x3x3
Max pooling 2x2 dropout 0.5
Flatten 5120

(e) Model V

Layer1 : Conv 8x3x3
Layer2 : Conv 16x3x3
Layer3 : Conv 16x3x3
Layer4 : Conv 32x3x3
Layer5 : Conv 32x3x3
Max pooling 2x2 dropout 0.5
Layer6 : Conv 64x3x3
Layer7 : Conv 64x3x3
Layer8 : Conv 96x3x3
Layer9 : Conv 96x3x3
Max pooling 2x2 dropout 0.5
Layer10 : Conv 128x3x3
Layer11 : Conv 256x3x3
Max pooling 2x2 dropout 0.5
Flatten 4894

(f) Model VI

図 4.2: 畳み込み層の数，フィルタ数の異なる 6 つの CNN モデル

例えば，図 4.2 (a) の Layer1 は，サイズが 3×3 である 16 枚のフィルタを持つ畳み込み層を表す．Max-Pooling は 6 つのモデル全てにおいて 2×2 のプーリングを計 3 度行

う。Flatten は 2 次元の特徴マップを 1 次元に変換することを表し、変換後のノード数は図 4.2 のそれぞれの通りである。以降は全てのモデルにおいて 512 のノードを持つ全結合層を 2 層通し、softmax 関数を出力層として使用することで、各クラスの事後確率を出力する。いずれのモデルにおいても、各層の活性化関数には ReLU を採用し、入力画像の正規化サイズを 50×200 、バッチサイズを 90、学習率を 10^{-4} とした。また本実験では、モデルの学習を安定化するために各畳み込み層の直後に Batch Normalization [36] を挿入した。これらの要素で構成される 6 つの CNN においては、いずれも 1000 エポックの学習を行った。

図 4.3 に 6 つのモデルにおける学習データ、検証データの Loss 値の推移グラフ、図 4.4 に学習、評価データの分類成功率の推移グラフをそれぞれ示す。損失関数は交差エントロピー誤差を採用し、最適化アルゴリズムとして Adam を用いた。また、検証サンプルの Loss 値 (図 4.3 (b)) は移動平均法によって、グラフの微小な増減を平滑化した。図 4.3 より、学習データと検証データの Loss 値がすべてのモデルで横ばいの状態になっており、これ以上学習の大きな進展が見込めないと推測できる。よって、これらの 6 つの異なる構造の CNN の性能をそれぞれ比較しても問題がないと判断することができる。

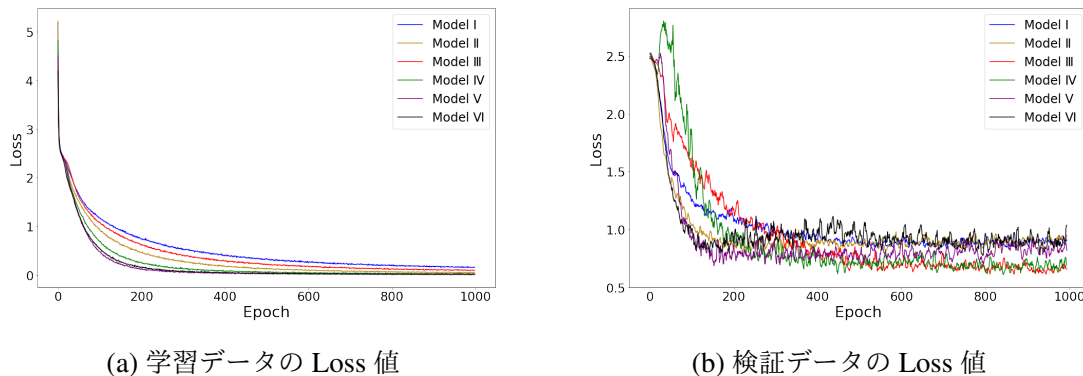


図 4.3: 6 つの CNN モデルにおける学習、検証データそれぞれの Loss 値

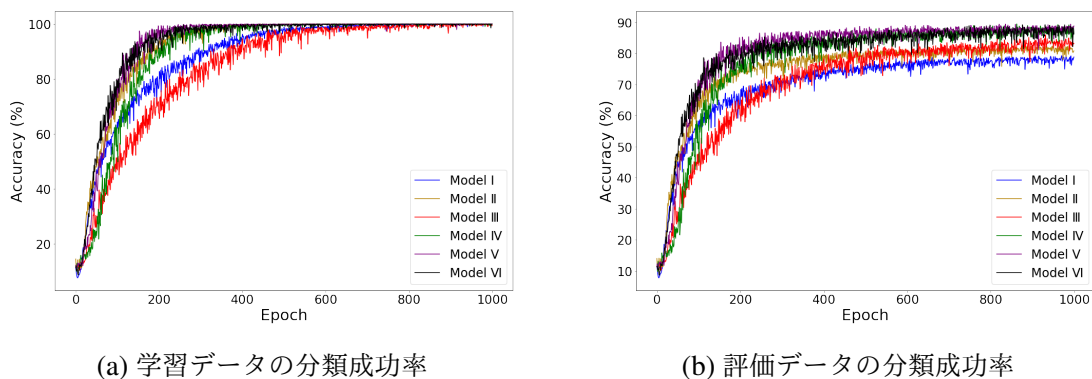


図 4.4: 6 つの CNN モデルにおける学習、評価データそれぞれの分類成功率

4.3 実験概要

前節で述べたデータセット，CNN モデルを用いて，本手法の有効性を確認する．任意の 1 つのハイパーパラメータを変更した数種類の CNN における各クラスの学習過程の比較，そして Reaction 値と交差エントロピー関数から得られた Loss 値との間で，最適エポック抽出に関する性能比較を行う．

4.3.1 ハイパーパラメータ変更による学習過程の比較

畳み込み層の数，畳み込み層のフィルター数それぞれを変更した時の学習エポックによる Reaction 値の推移を，CNN モデルごとに各クラスで比較する．また，学習画像，評価画像それぞれの分類成功率の推移グラフや，クラスごとの損失グラフも同様に比較することで，各クラスにおけるハイパーパラメータへの依存性も調査を行う．

畳み込み層の数の比較用として用いる CNN モデルは，前節の図 4.2 に示す，Model II，Model IV，Model V，Model VI の 4 種類のモデルを使用する．また，畳み込み層のフィルター枚数の比較用として用いる CNN モデルは，Model I，Model II，Model III，Model IV の 4 種類のモデルを使用する．

畳み込み層の数，畳み込み層のフィルター数の 2 つのパラメータを比較の対象とし，構造の似た CNN モデルの間で各クラスの学習過程にどのような違いがあるかを確認する．

4.3.2 Reaction 値，Loss 値を用いた最適エポックの抽出

ニューラルネットワークの汎化性能を高める学習テクニックの 1 つに Early Stopping がある．これはエポックが更新される度に，損失関数から得られた値を随時記録することで，損失値が予め定めた閾値を下回ったとき，そのエポックで学習を中断する技術である．本実験では，学習データから得られた Reaction 値と検証データから得られた Loss 値を用いて，前後のエポックの変動からエポックの学習収束判定を行う．エポック 1 から順に走査を始め，下記に示す式 (4.1)，(4.2) の条件を満たしたときに学習完了とみなし，その時点のエポックを学習の最適エポックとする．ここで， i は走査時のエポック， V は Reaction または Loss の値， S は走査時の Reaction 値または Loss 値の標準偏差である．ここで両標準偏差は，走査時のエポックから 100 エポック前までの 100 個の値から算出する．また，走査時のエポックが学習完了か否かを判定する際の閾値 (T) を設定する．この閾値が小さいほど，エポック間の差分が微小でないと収束判定されない条件式となっている．この閾値を 0 から 1 の間で変化させながら，Reaction 値，Loss 値それぞれで収束

条件を満たすエポックを抽出し，収束判定性能を比較する．

$$(V_i - V_{i-1})/S_i < T \quad (0 < T \leq 1) \quad (4.1)$$

$$V_{i-1} > V_i \quad (i > 1) \quad (4.2)$$

さらに，本実験では各 CNN モデルを予め 1000 エポック学習した後に記録データを元に最適エポックを求めるため，Reaction 値が最小となるエポックと Loss 値が最小となるエポックにおける 13 クラス全体の分類成功率も比較する．この比較によって，学習完了後の最適エポックの抽出法として Reaction 値が有効であるかどうかの検証も行う．

第 5 章

結果と考察

5.1 ハイパーパラメータの違いによる学習過程分析実験

畳み込み層の数の違いによる学習過程の分析実験では，比較用の Model II, Model I V, Model V, Model VI をそれぞれ Model 3layers, Model 6layers, Model 8layers, Model 11layers と称して実験を行う。

また，畳み込み層のフィルター数の違いによる学習過程の分析実験では，比較用の Model I, Model II, Model III, Model IV をそれぞれ Model 3layers-64filters, Model 3layers-128filters, Model 6layers-64filters, Model 6layers-128filters と称して実験を行う。

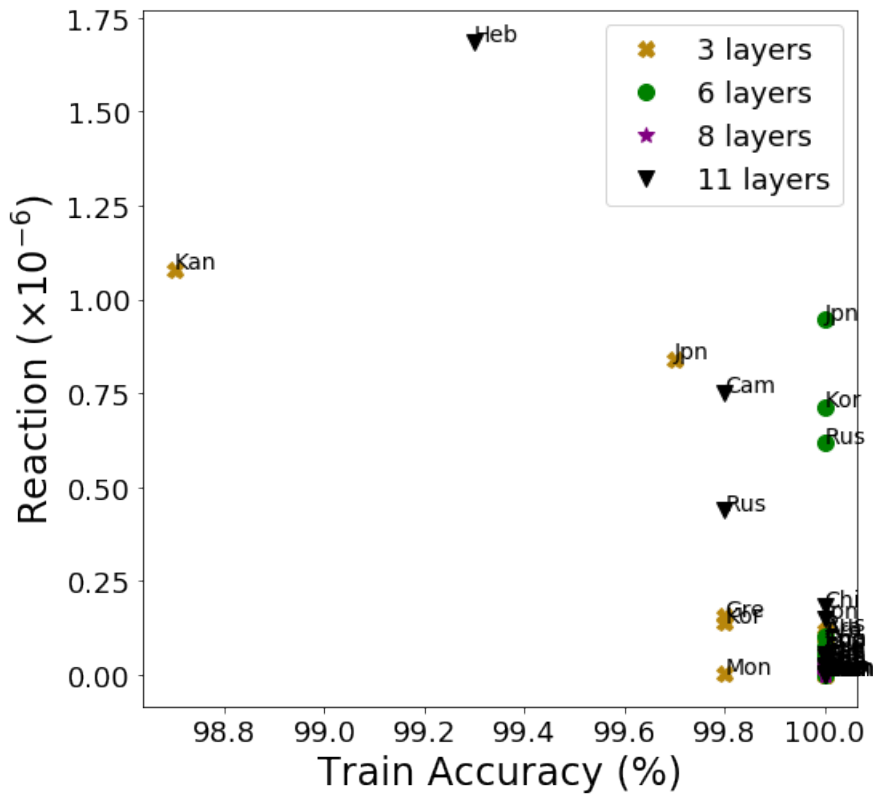
5.1.1 畳み込み層の数の違い

まず最初に，4つのモデルにおいて 1000 エポック目の各言語の Reaction 値と分類成功率の散布図を図 5.1 に示す。図 5.1 (a) は学習に用いた画像を分類した時の Reaction 値と分類成功率の散布図，図 5.1 (b) は未学習である評価画像を分類した時の Reaction 値と分類成功率の散布図である。図 5.1 (a) より，学習画像の精度が高い言語，すなわち学習が十分に進んでいる言語は Reaction 値が低くなり全体的にプロットが右下部分に集中していることがわかる。また Model 3layers のカンナダ語 (Kan)，日本語，Model 6layers の日本語 (Jpn)，韓国語 (Kor)，ロシア語 (Rus)，Model 11layers のカンボジア語 (Cam)，ヘブライ語 (Heb)，ロシア語のような，図 5.1 (a) において Reaction 値が比較的高い言語は，評価サンプルを用いた分類成功率が比較的低いことが図 5.1 (b) からわかる。モデルによって Reaction 値や分類成功率が異なる言語が多数存在することから，層の数のみが異なる CNN モデルでは，各クラスにおいて学習過程にばらつきがあったことが考えられる。

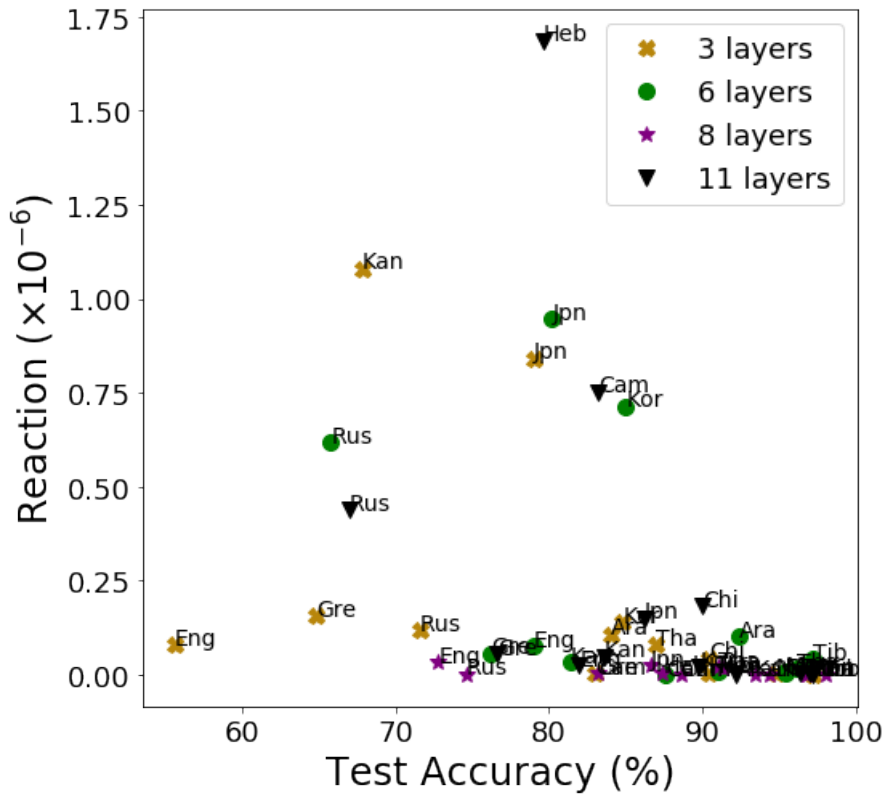
次に，図 5.2 に，英語クラスの Reaction 値の推移，Loss 値の推移，学習データ，評価データの分類成功率の推移グラフをそれぞれ示す。本実験では，グラフの折れ線を平滑

化して推移を俯瞰できるようにするため、4種のグラフすべてに対して移動平均法を用いて各値を変換した。残りの12言語の分析結果は、付録として本論文の末尾に掲載する。図5.2(a)に示す各言語のReaction値の推移グラフから、学習エポックが進むにつれ、一度単調増加をした後、単調減少する傾向が4つのモデルから読み取ることができる。Reaction値はCNNの特徴マップにおける層間の勾配値を表す値であるため、Reaction値の増加は、特徴マップに微小変化を加えたときの各ニューロンの重みの変化が大きいことを指す。そのため、Reaction値がピークに達した付近のエポックが、CNNの学習が進む際の、最も学習が活性化したエポックであると考えられる。したがって、学習データ(図5.2(c))、評価データ(図5.2(d))それぞれの分類成功率は、Reaction値のピーク付近のエポックにおいて急上昇している。Loss値の推移グラフ(図5.2(b))は、エポックが進むにつれ単調減少が続くだけであるので、学習が活性化したエポックが明らかではなく、モデル同士の比較も難しいが、Reaction値の推移グラフではグラフの形が山なりになっているので、ニューロンが活性化したエポックがどの時点での学習であるかが明らかである。

また、図5.2(a)、図5.2(d)から、4つのモデルにおけるReactionの推移を比較すると、より早い段階のエポックでReaction値がピークに達したモデルは他のモデルと比べて、1000エポック時点での評価サンプルの分類成功率が高いことがわかる。Reaction値が早い段階で増加し、素早く減少方向に向かうことは、学習が進むことでニューロンの重みの変化が徐々に小さくなり、分類に最適な学習が上手くできていることを示す。そのため、Reaction値の山なりが尖っていて、ピーク部分が明確であるモデルはそのクラスに対して学習が安定しており、高い精度を得たと考えられる。

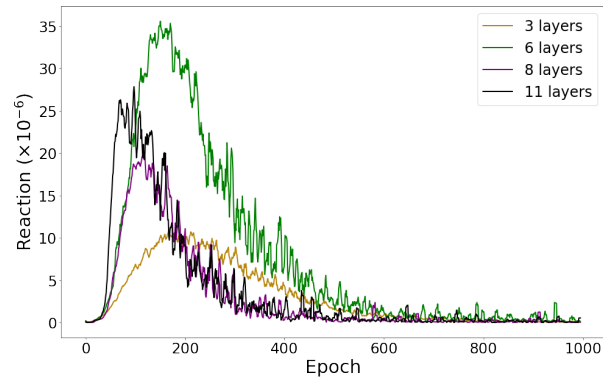


(a) 学習データ

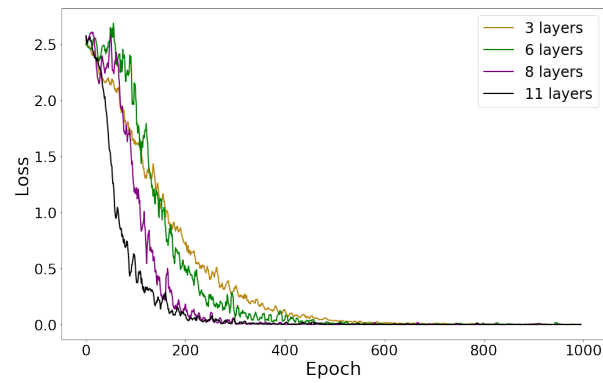


(b) 評価データ

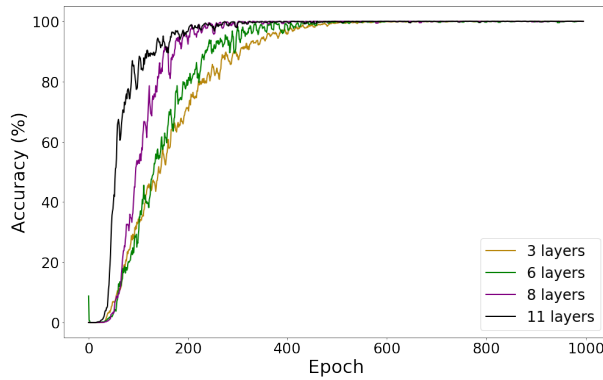
図 5.1: SIW-13 による Reaction 値と分類成功率の散布図



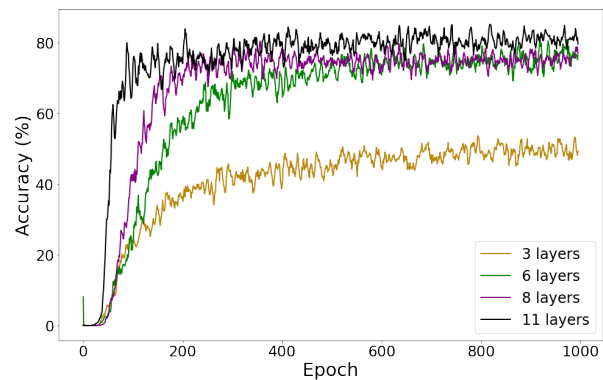
(a) Reaction 値の推移



(b) Loss 値の推移



(c) 学習データの分類成功率の推移



(d) 評価データの分類成功率の推移

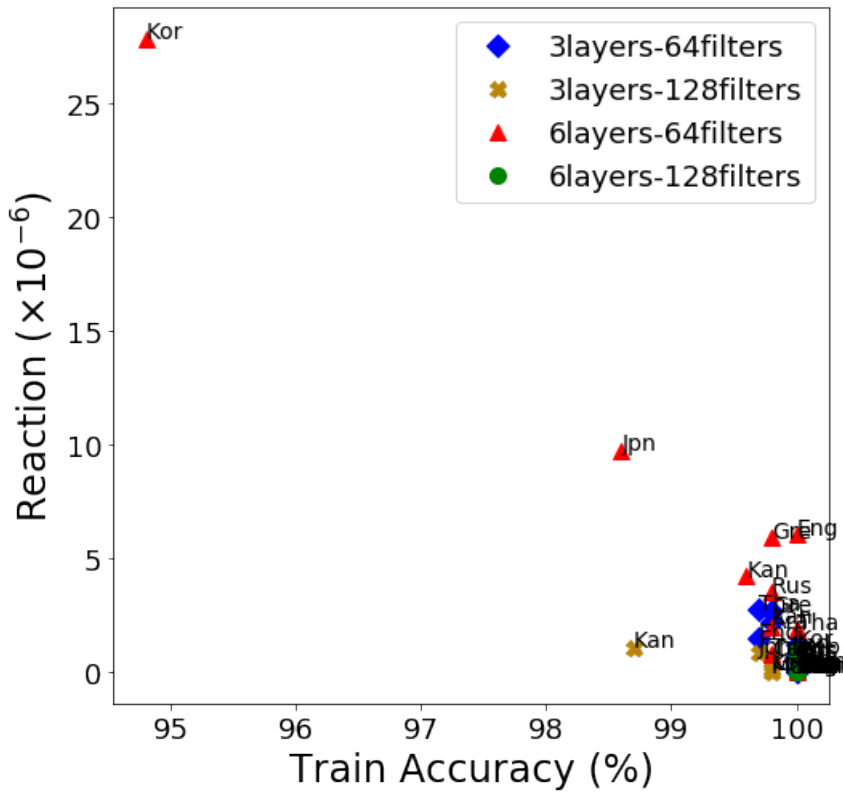
図 5.2: 英語クラスの実験結果

5.1.2 畳み込み層のフィルター数の違い

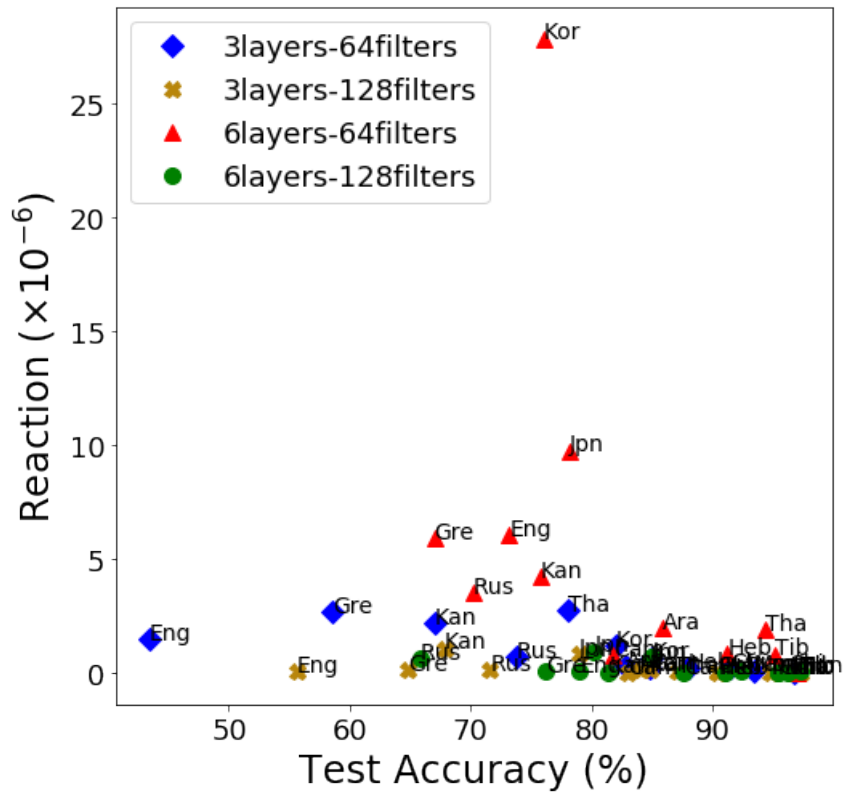
前節の実験と同様に、4つのモデルにおいて、1000エポックの学習更新時の各言語の Reaction 値と分類成功率の散布図を図 5.3 に示す。図 5.3 より、Model 6layers-64filters では他のモデルよりも、学習、評価画像双方において高い Reaction 値をもつ言語があることがわかる。また、Model 6layers-128filters のすべての言語が散布図の右下部分に集中しており、図 5.3 (b) については、いずれの言語においても、他のモデルより分類成功率が高く、Reaction 値も比較的 0 に近いことから、Model 6layers-128filters が 4 つのモデルの中で最も上手く学習が進んでいるモデルであると判断できる。この結果に対し、Model 6layers-64filters は、畳み込み層の数が Model 6layers-128filters と同じであるにも関わらず、各層のフィルター数の不足が起因して、Reaction 値のばらつきが生じたと考えられる。

次に、図 5.4 ~ 5.8 に、中国語、英語、ギリシャ語、ロシア語、チベット語クラスのそれぞれの Reaction 値の推移、Loss 値の推移、学習サンプルおよび評価サンプルの分類成功率を示す。これらのグラフは、前節と同様に移動平均法を用いて値の変動を平滑化した推移グラフである。残りの 7 言語の分析結果は本論文の付録として、末尾に掲載する。図 5.4 (a)、図 5.8 (a) のように、Reaction 値のピークが早い段階のエポックで訪れ、その後の Reaction 値の減少が急である言語は、図 5.4 (d)、図 5.8 (d) に示すように、他の言語と比べ高い分類成功率が得られていることがわかる。

一方で、図 5.5 (a)、図 5.6 (a)、図 5.7 (a) のような、Reaction 値のピークが他より遅いエポックで訪れるモデルや、ピーク後に Reaction 値が 0 に近付いていないモデル、あるいは隣り合うエポック間で Reaction 値の変動が常に激しいモデルは、分類成功率が低い結果となった。よって、それらの推移を示す言語は学習過程の不安定な言語であると判断することができる。図 5.4 (b) ~ 5.8 (b) においても、言語やモデルによって Loss 値の推移はそれぞれ異なることがわかる。しかし、Reaction 値を用いた分析では、モデルや言語ごとの学習過程の比較が Loss 値の推移グラフと比べて容易であり、より多くの情報をもとに分析することが可能である。また 13 言語のいずれも、学習過程がモデルによって異なることから、CNN の学習性能がハイパーパラメータの違いに依存することが、Reaction 値によって、より効果的に確認できた。



(a) 学習データ



(b) 評価データ

図 5.3: SIW-13 による Reaction 値と分類成功率の散布図

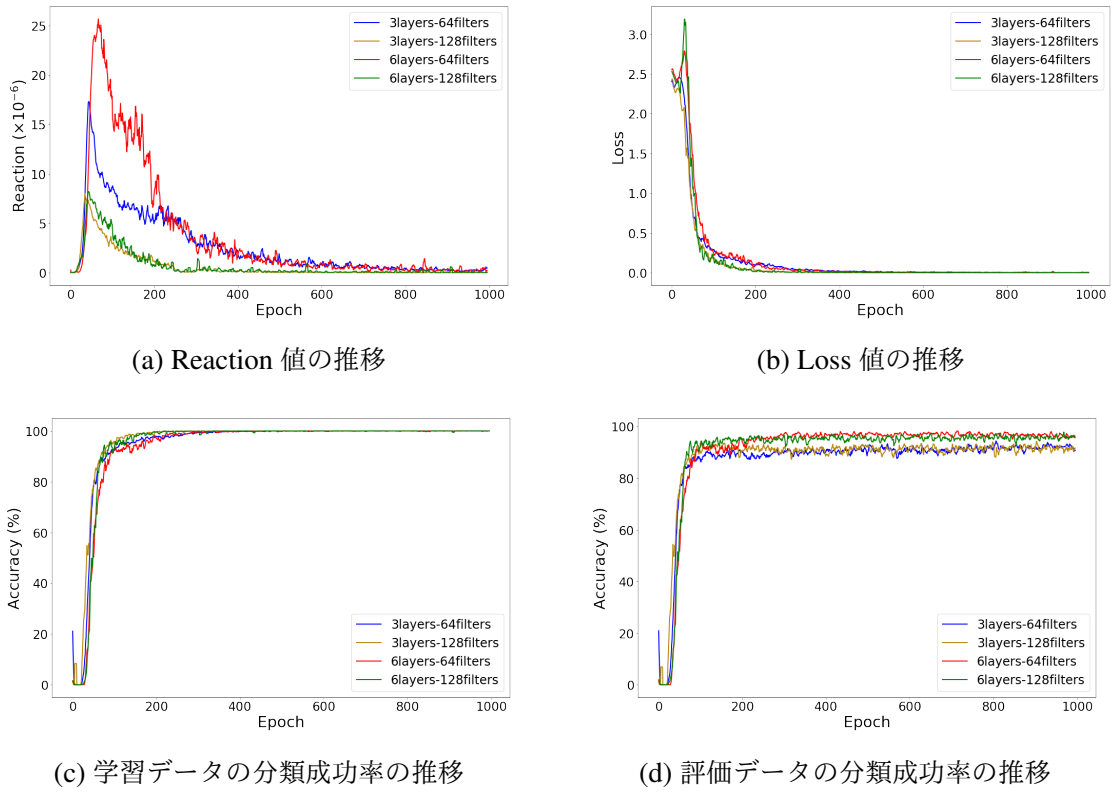


図 5.4: 中国語クラスの分析結果

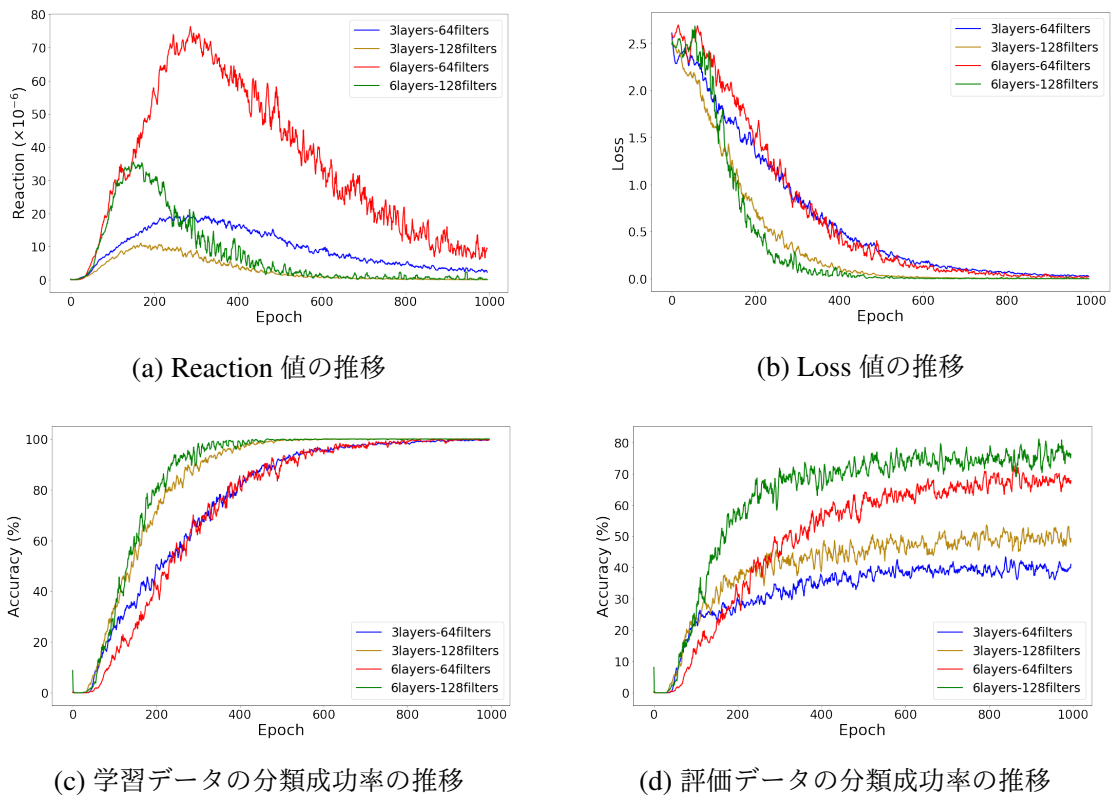


図 5.5: 英語クラスの分析結果

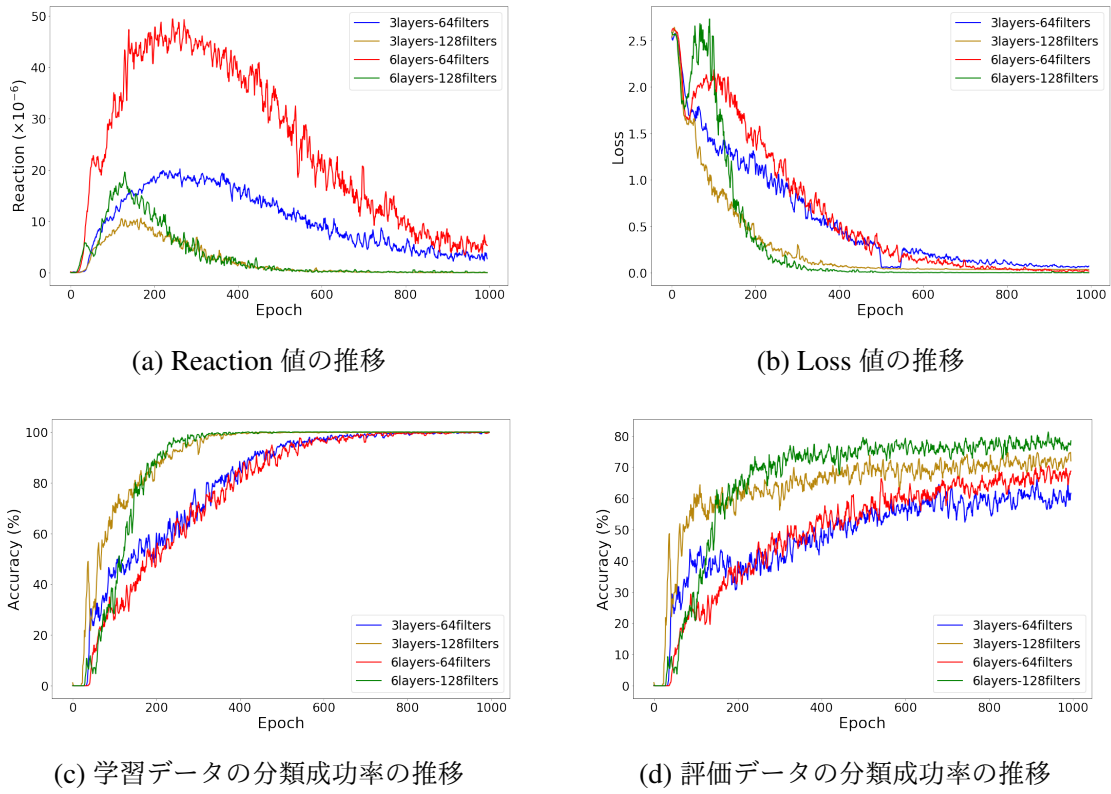


図 5.6: ギリシャ語クラスの分析結果

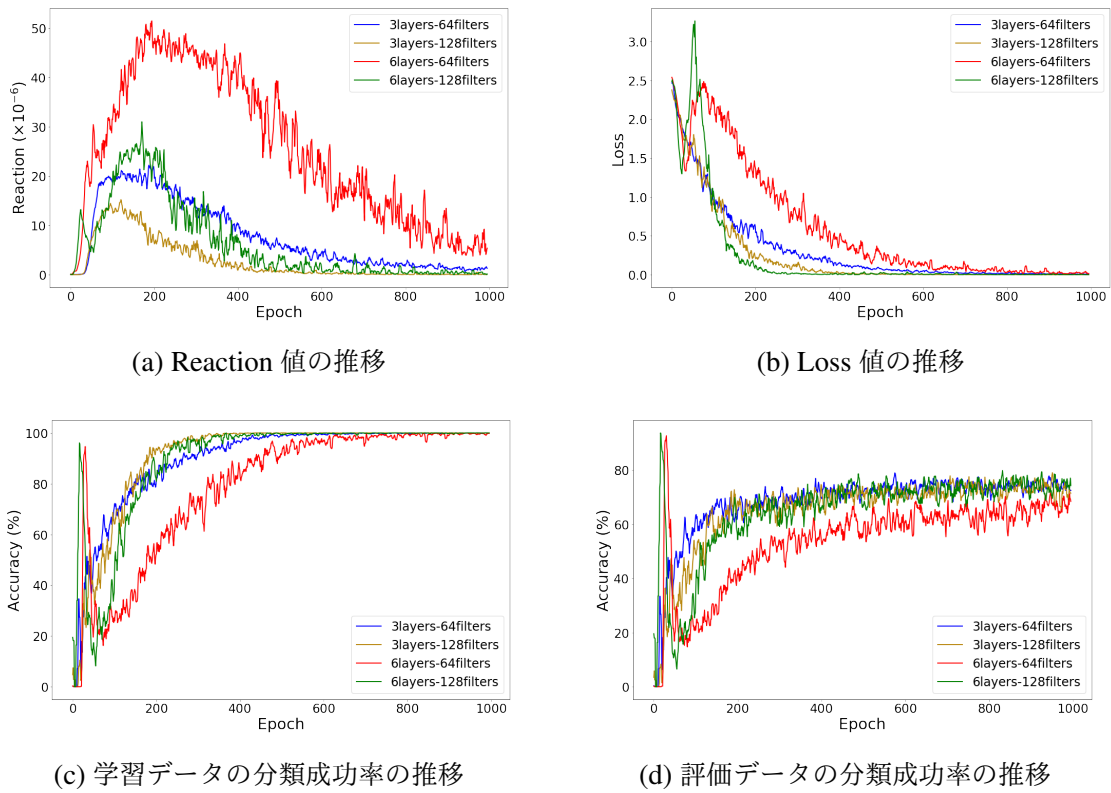
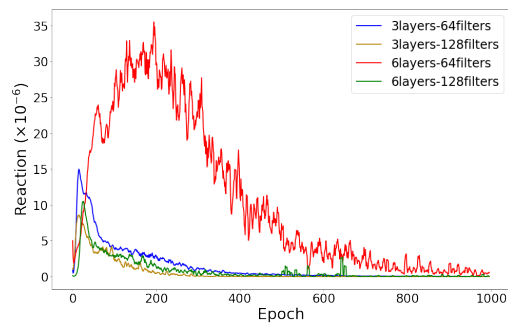
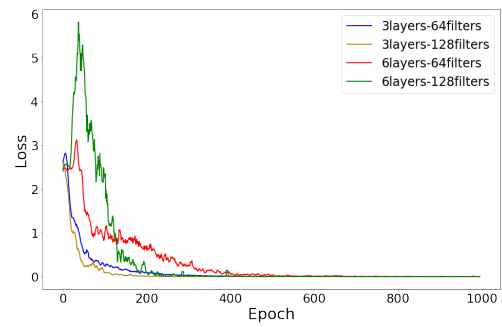


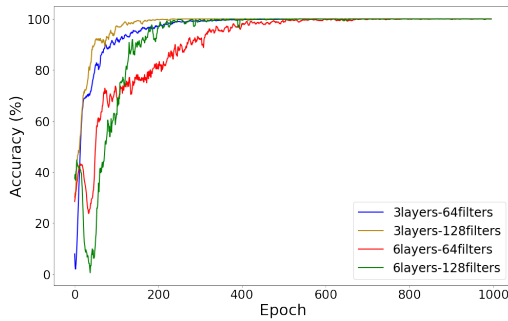
図 5.7: ロシア語クラスの分析結果



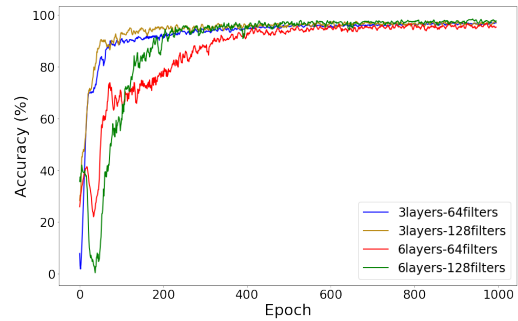
(a) Reaction 値の推移



(b) Loss 値の推移



(c) 学習データの分類成功率の推移



(d) 評価データの分類成功率の推移

図 5.8: チベット語クラスの分析結果

5.2 最適エポック抽出実験

Reaction 値と Loss 値それぞれにおいて、4.3.2 節の条件式を満たしたエポックを抽出する方法と、1000 エポック中で Reaction 値, Loss 値がそれぞれ最小であるエポックを抽出する方法の 2 種類で、収束判定のためのエポック抽出の汎化性能を評価する。ここで、6 つの CNN モデルによる全 13 言語の学習データから算出した、Reaction 値の平均値の推移グラフをそれぞれ図 5.9 に示す。

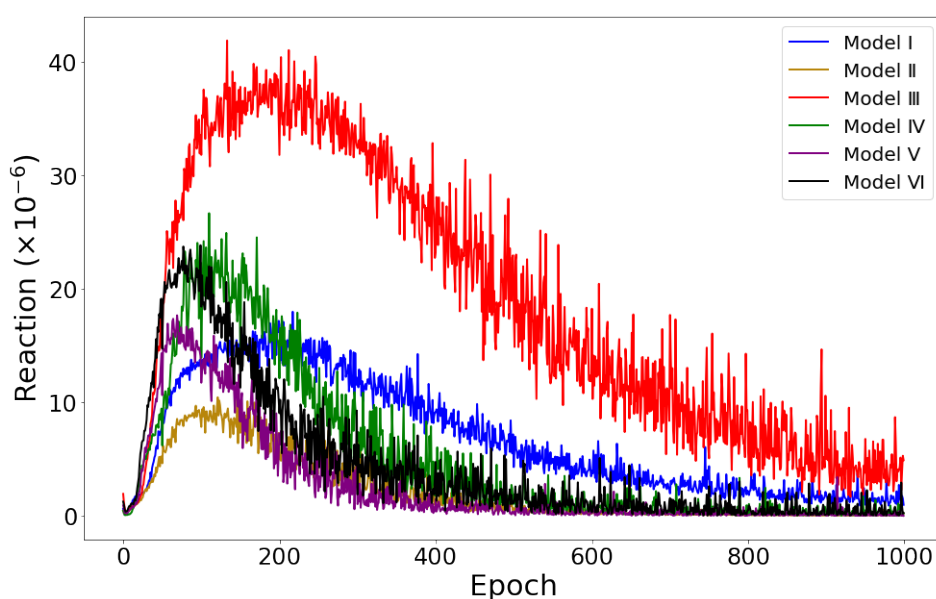


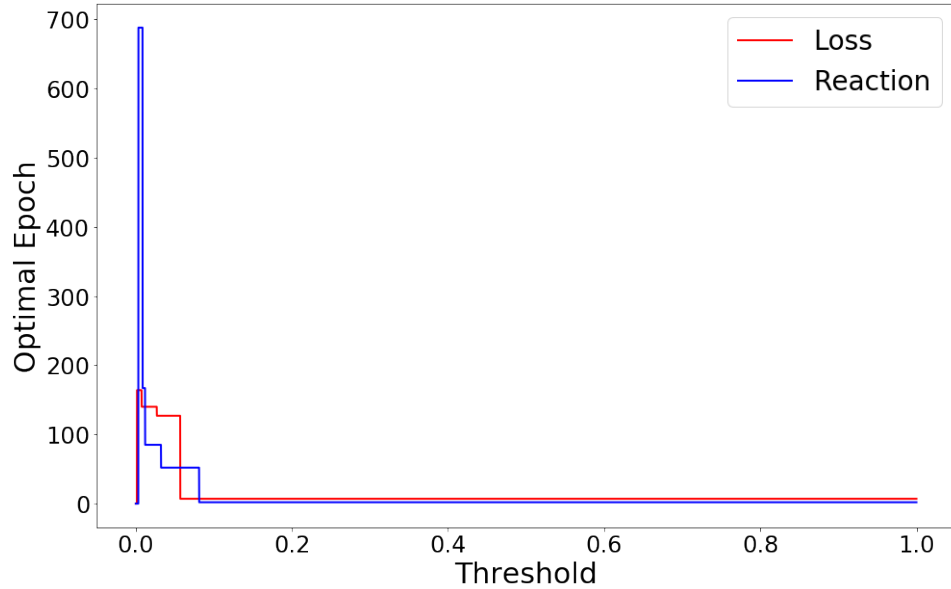
図 5.9: 6 つの CNN モデルの Reaction 値の推移

5.2.1 学習収束判定の性能比較

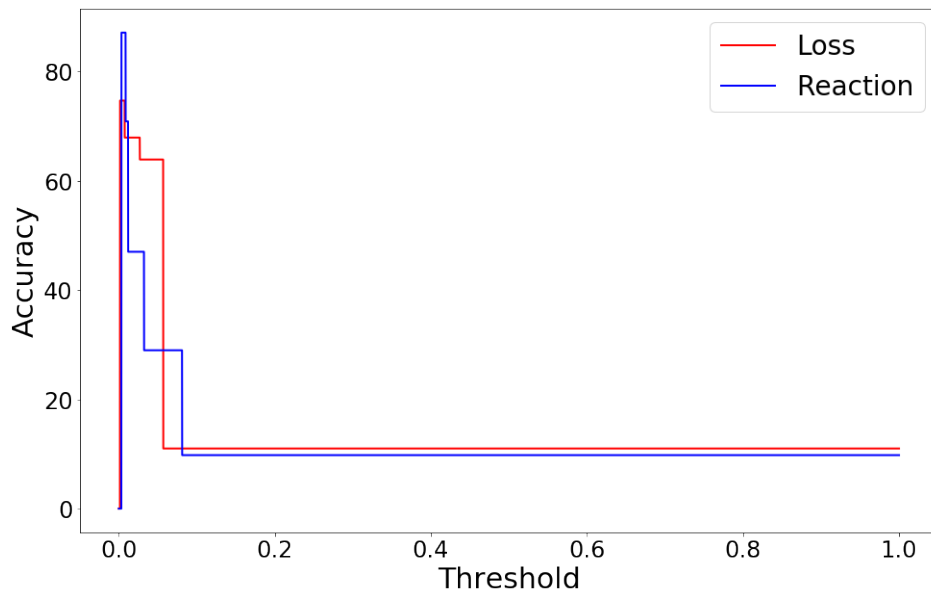
図 5.9 に示した Reaction 値の推移と 4.2 節の図 4.3 (a) で示した Loss の推移から、閾値を変更することで先述した式の条件に当てはまり収束判定されたエポック (図 5.10 (a)) と、そのエポックにおける 13 言語の分類成功率を表した結果 (図 5.10 (b)) を示す。比較に使用した CNN モデルは、前章の図 4.2 (d) に示す Model IV を使用した。図 5.10 (b) より、収束判定をした時に抽出されるエポックにおける分類成功率の最大値は、Loss 値を用いた時より、Reaction 値を用いた方が高くなった。

また図 5.11 は、先程の Model IV と同様に、Model I, Model II, Model III, Model V, Model VI において同様の比較実験を行った時のそれぞれの分類成功率を表した結果である。図 5.11 より、Model V 以外の CNN モデルにおいて、Reaction 値を用いた収束判定は、Loss 値を用いたときより、高い分類成功率をもつ学習エポックを抽出できた。閾値を

変更することで、Loss 値を用いた収束判定のほうが高い分類成功率を得る場合もあるが、閾値の設定次第では、Reaction 値は収束判定により有効であることがわかる。

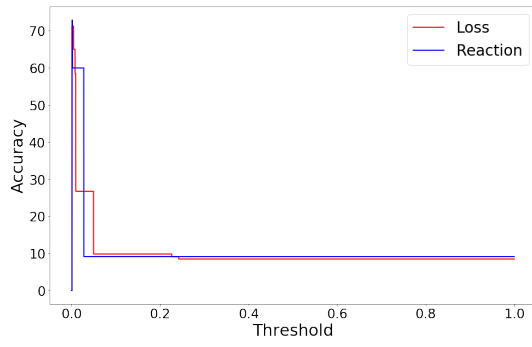


(a) 各閾値による最適エポック

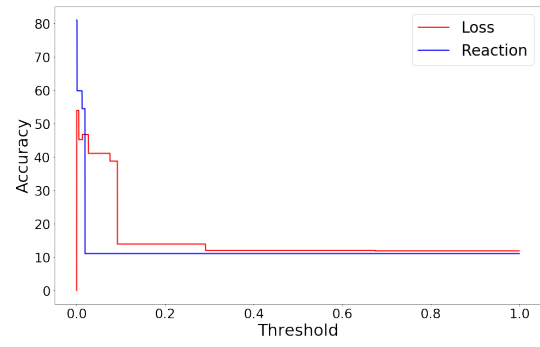


(b) 各最適エポックによる分類成功率

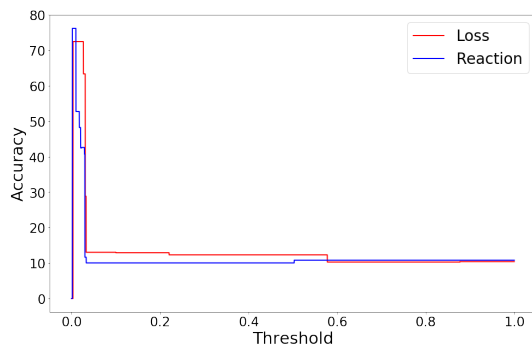
図 5.10: 閾値の変動による Loss, Reaction 値を用いた収束判定の結果



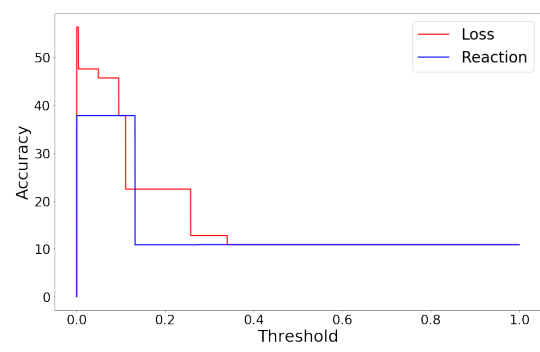
(a) Model I



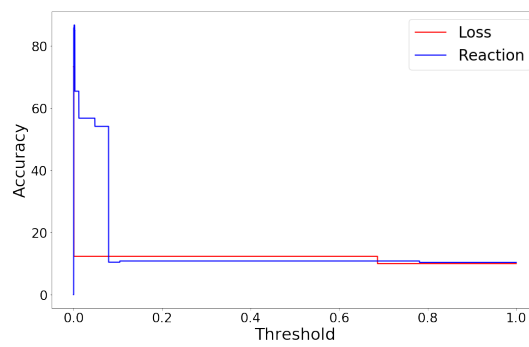
(b) Model II



(c) Model III



(d) Model V



(e) Model VI

図 5.11: 各 CNN モデルで収束判定を用いて得られたエポックの分類成功率

5.2.2 最小値による分析成功率の比較

4.2 節の図 4.3 (a) に示す検証データにおける Loss 値の推移と、図 5.9 に示す学習データにおける Reaction 値の推移において、各モデルで最も値が小さいエポックと、それらのエポックでの各分類成功率を比較した結果を表 5.1 に示す。Reaction 値は 1000 エポックの中で最高値をもつエポック以降で、最小値をもつエポックを抽出した。表 5.1 から、6 つの CNN モデルすべてにおいて Loss 値を用いて得られたエポックより高い成功率をもつエポックを得ることに成功したことがわかる。

これらの結果より、本手法で扱う Reaction 値を用いた収束判定による学習エポックの抽出が Loss 値を用いた抽出よりも有効であることがわかる。また、Reaction 値は検証データを用いることなく、学習データのみで学習過程の分析を効率よく行えることが確認できた。

表 5.1: 最小 Loss 値, 最小 Reaction 値をもつエポックにおける分類成功率

Model	Loss		Reaction	
	Epoch	Accuracy	Epoch	Accuracy
I	517	77.83	980	78.98
II	321	79.21	948	82.37
III	931	85.94	931	85.94
IV	919	88.85	987	89.17
V	154	81.72	991	89.43
VI	189	82.78	748	88.92

第 6 章

結言

6.1 まとめ

本研究では，損失関数に代わる CNN の学習進捗状況を分析するための手法を提案した．CNN を用いた分類の際に判断根拠を可視化する Grad-CAM を応用することで，入力画像の学習の活性度を統計的に算出し，その値を“Reaction 値”と定義した．Reaction 値によって各クラスの学習の進捗度合いの算出が可能となり，自作の 6 つの CNN を用いることで，言語判定における CNN の学習過程の分析を行った．

言語判定における各 CNN の学習過程の分析実験では，学習エポックによる Reaction 値の推移を CNN モデルごとに各クラスで算出することで比較を行った．実験の結果，サンプルを安定して学習できるかどうかは，畳み込み層の数や層のフィルター数といった，ハイパーパラメータに依存することが示された．また，これらのパラメータの値が学習サンプル数に対して十分である場合，各クラスで学習が特に活性化したエポックを明らかにすることができた．一方で，パラメータ値が不十分である場合，学習過程が不安定になるクラスが発生し，結果としてモデルの性能にも影響をすることも確認できた．

Loss 値との性能比較実験では，過学習を防ぐために利用される Early Stopping に倣って，閾値や最小値を用いた収束判定実験を行った．実験の結果，Reaction 値を用いた収束判定は，Loss 値を用いたときより分類成功率の高いエポックが得られ，優位性を確認することができた．

6.2 今後の課題

近年ニューラルネットワークに関する研究として、学習の効率化を目指した手法が数多く提案され、その優位性が確認されている。そのため本研究でも、提案した“Reaction 値”の欠点を改善し、さらなる優位性を実証する必要がある。

本実験では、閾値を変更することで、Loss 値を用いた収束判定よりも性能の良い収束判定を確認できたが、毎度閾値を変える方法は実用的ではない。そのため、Reaction 値を用いた収束判定が一般の問題に応用できるために、閾値の最適決定法を提案する必要がある。

また表 6.1 に、本実験で用いた SIW-13 内の学習データにおける、6 つの CNN モデルの Loss 値、Reaction 値の 1 エポック分の計算時間をそれぞれ示す。本実験で比較した Loss 値の計算時間と大きな差はないが、Reaction 値は CNN の最終畳み込み層を用いて値を算出するため、使用する CNN の層が複雑になるほど計算速度が遅くなる。また学習に用いる画像を 1 枚ずつ計算することで、クラスごとの学習の活性度を算出するため、扱う学習サンプルの規模によって計算速度が大きく異なる。本研究で用いた CNN モデルの中で最も多層である Model VI では、Reaction 値の 1 エポック分の計算時間が約 69 秒であった。そのため、扱う CNN 構造の複雑さやサンプルの規模に頑健な処理を可能にするアルゴリズムの検討が課題となる。

また本論文では、言語判定問題に着目して CNN の学習進捗状況の分析を行ったが、言語判定問題のみならず他の分類問題に対しても、本手法が有効であるかどうかを検証する必要がある。

表 6.1: 各モデルにおける Loss 値, Reaction 値の計算時間 (秒)

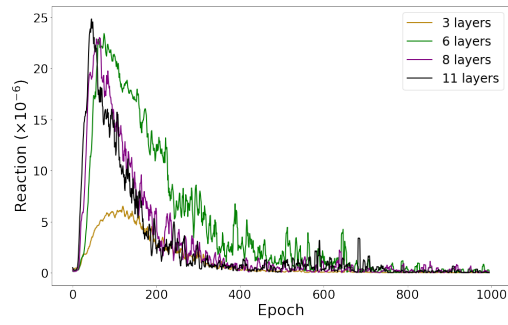
Model	I	II	III	IV	V	VI
Loss	46	48	49	50	55	60
Reaction	44	44	50	52	55	69

付録 A

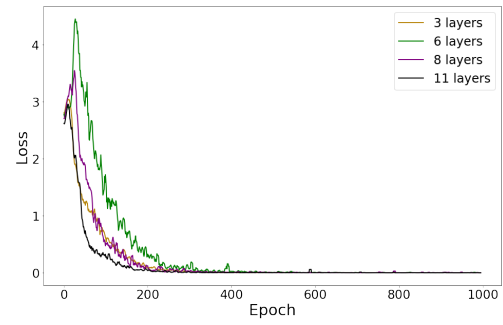
各モデルにおける言語ごとの分析 結果

5.1 節で示した 2 種類の分析実験の言語ごとの結果で，省略した言語の分析結果を付録として掲載する。

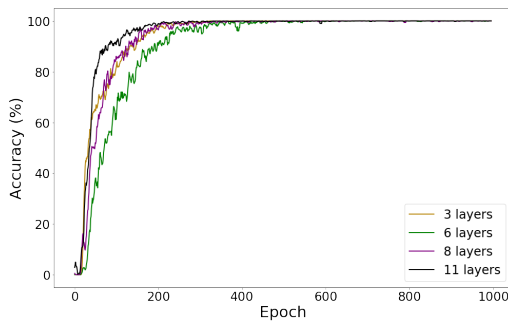
A.1 畳み込み層の数の変更による分析結果



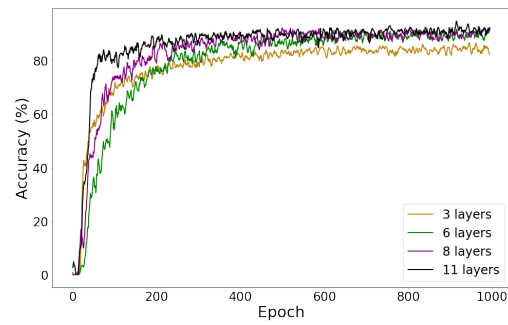
(a) Reaction 値の推移



(b) Loss 値の推移

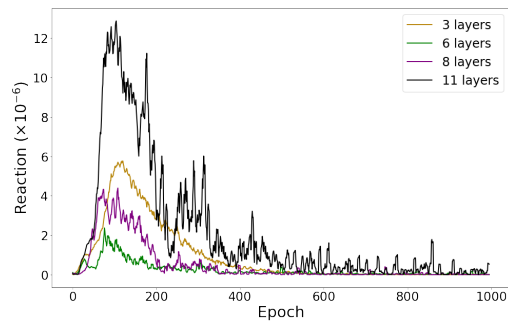


(c) 学習データの分類成功率の推移

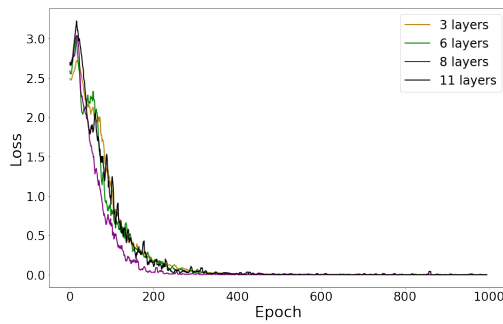


(d) 評価データの分類成功率の推移

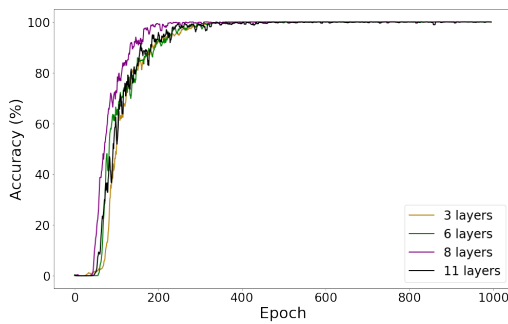
図 A.1: アラビア語クラスの分析結果



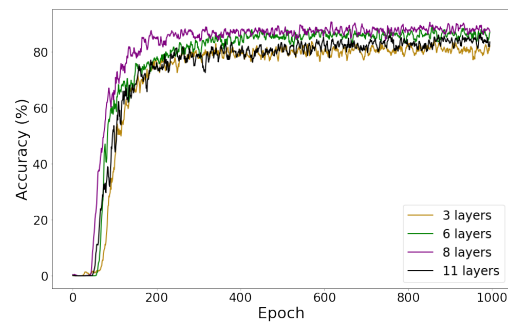
(a) Reaction 値の推移



(b) Loss 値の推移

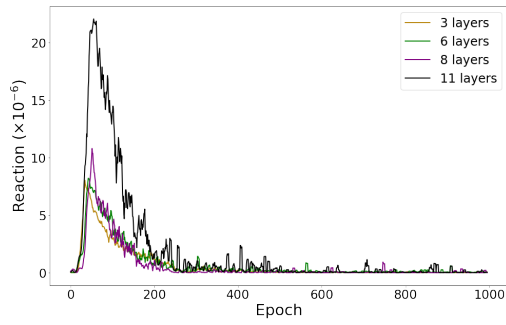


(c) 学習データの分類成功率の推移

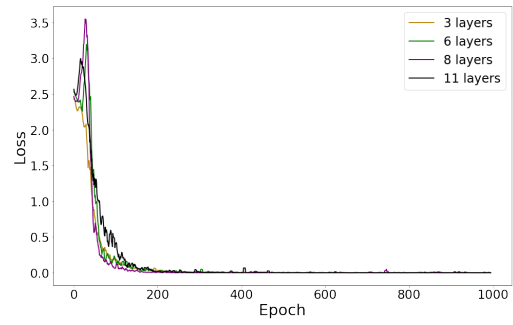


(d) 評価データの分類成功率の推移

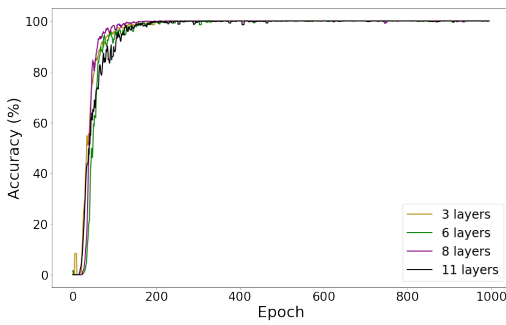
図 A.2: カンボジア語クラスの分析結果



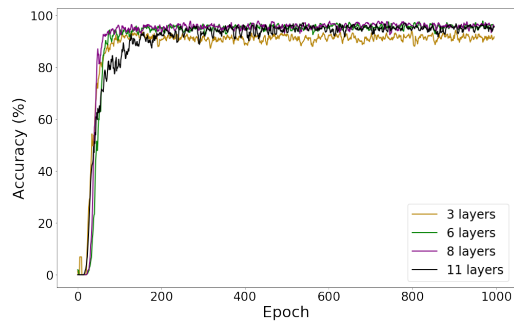
(a) Reaction 値の推移



(b) Loss 値の推移

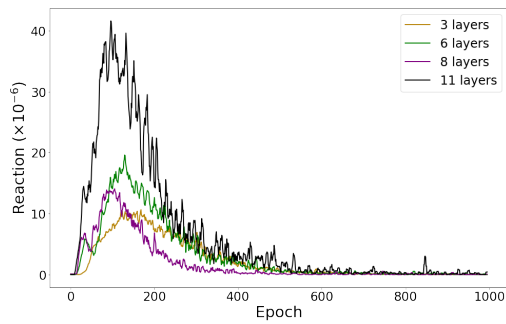


(c) 学習データの分類成功率の推移

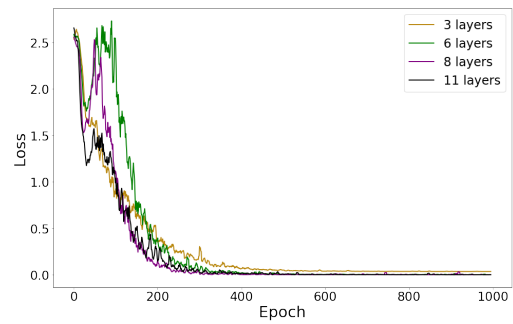


(d) 評価データの分類成功率の推移

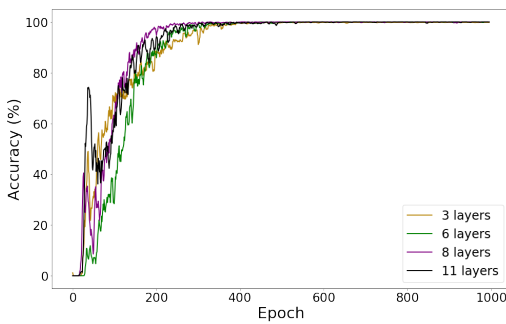
図 A.3: 中国語クラスの分析結果



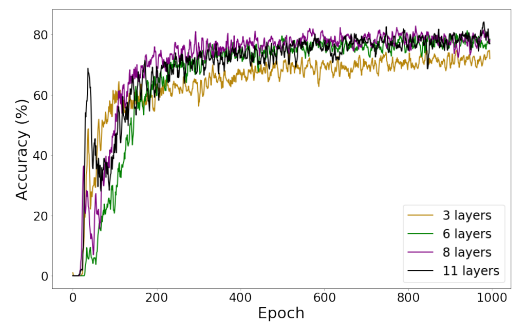
(a) Reaction 値の推移



(b) Loss 値の推移

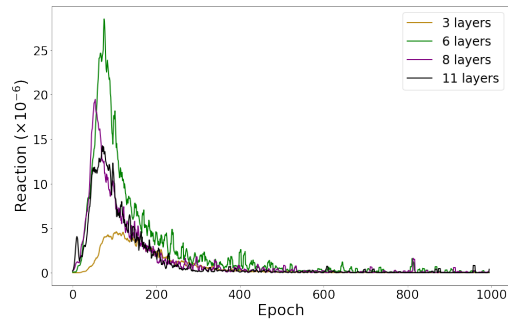


(c) 学習データの分類成功率の推移

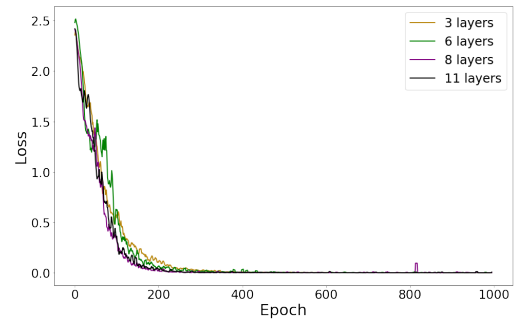


(d) 評価データの分類成功率の推移

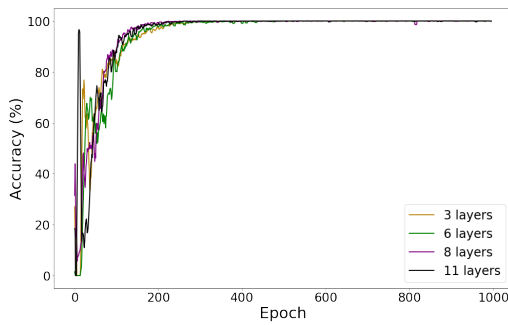
図 A.4: ギリシャ語クラスの分析結果



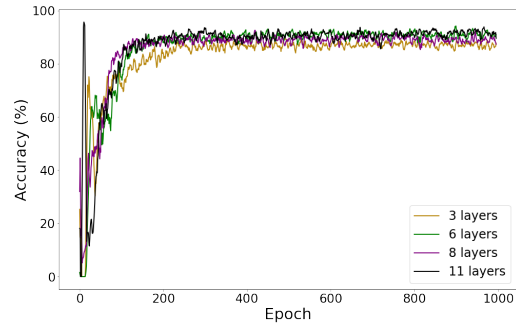
(a) Reaction 値の推移



(b) Loss 値の推移

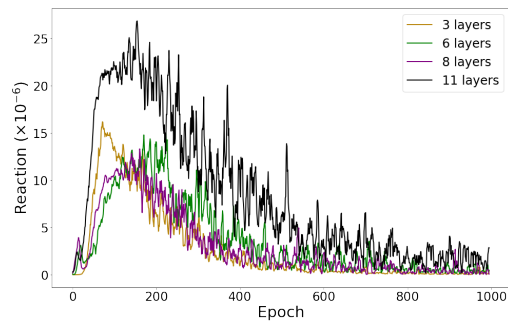


(c) 学習データの分類成功率の推移

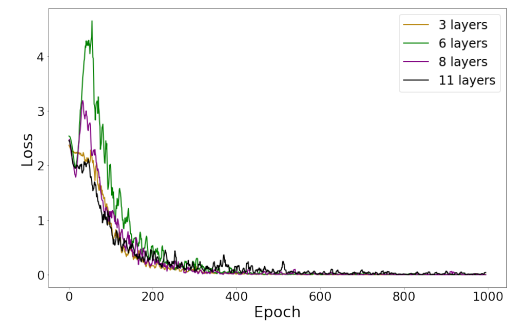


(d) 評価データの分類成功率の推移

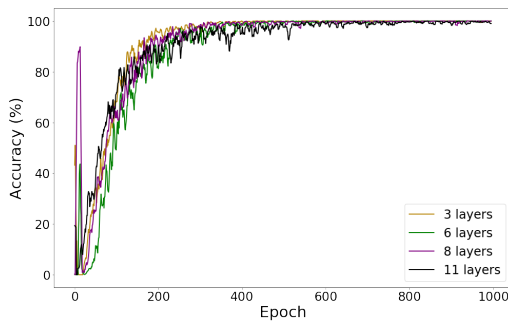
図 A.5: ヘブライ語クラスの分析結果



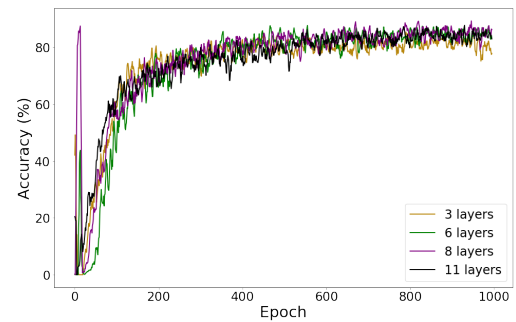
(a) Reaction 値の推移



(b) Loss 値の推移



(c) 学習データの分類成功率の推移



(d) 評価データの分類成功率の推移

図 A.6: 日本語クラスの分析結果

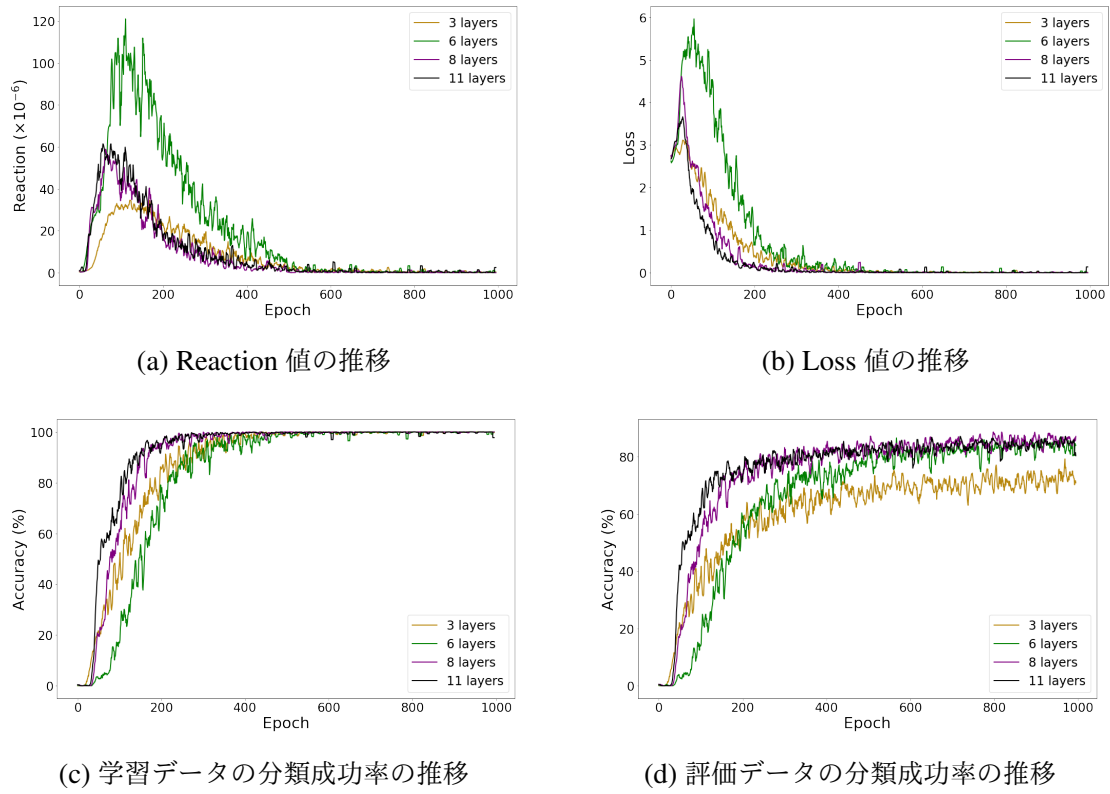


図 A.7: カンナダ語クラスの分析結果

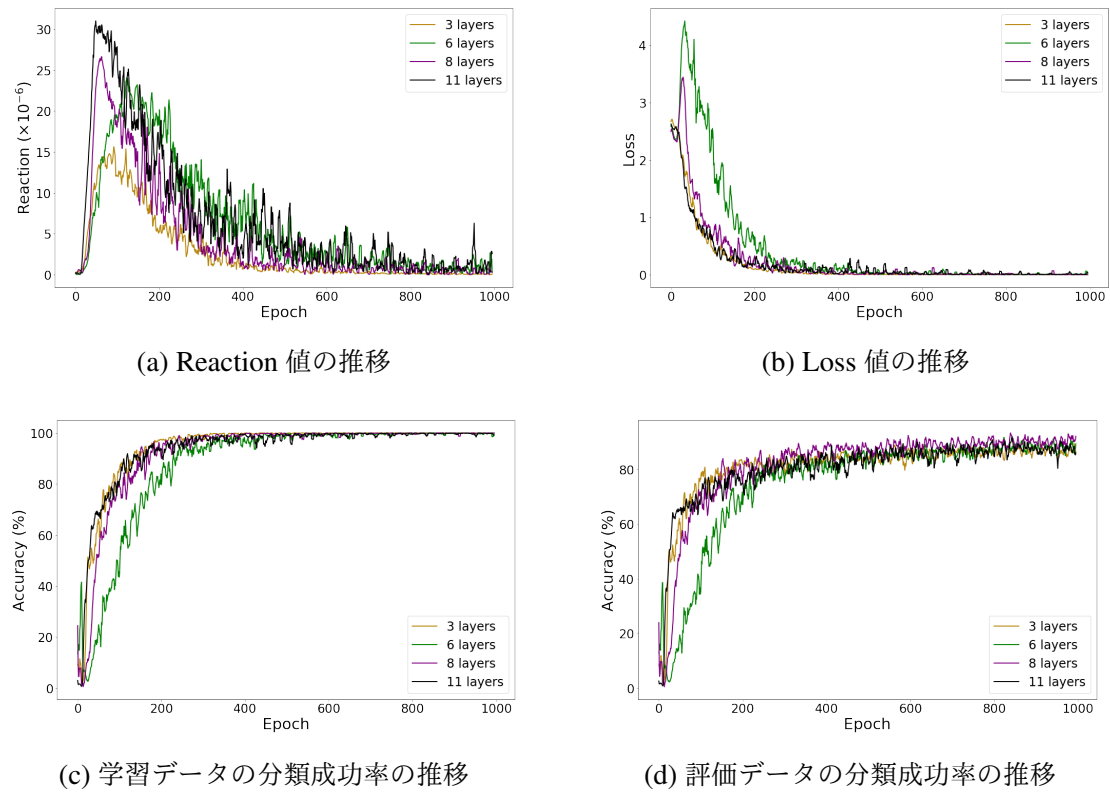
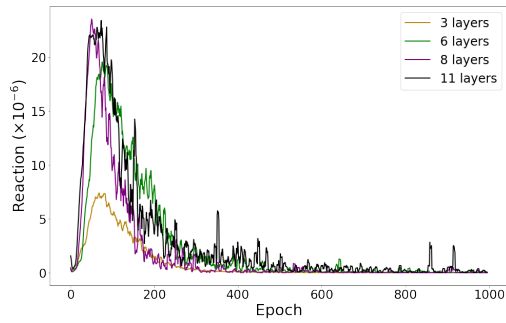
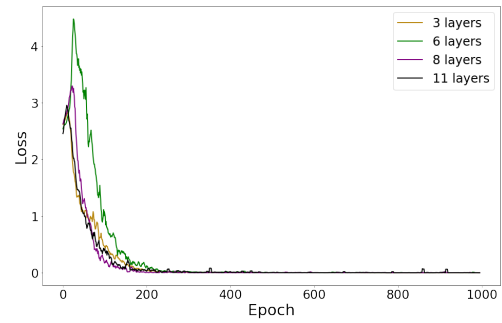


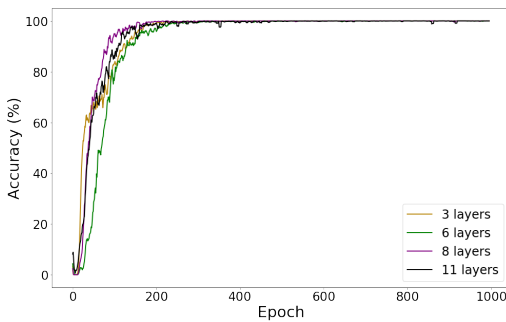
図 A.8: 韓国語クラスの分析結果



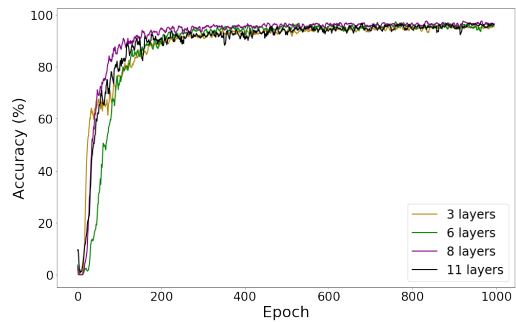
(a) Reaction 値の推移



(b) Loss 値の推移

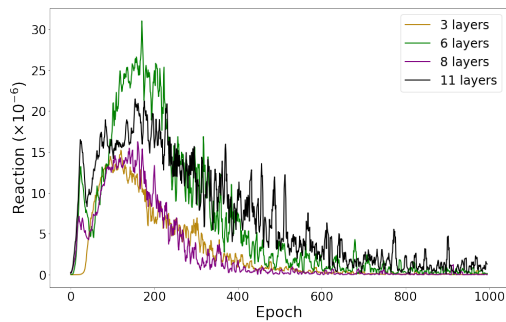


(c) 学習データの分類成功率の推移

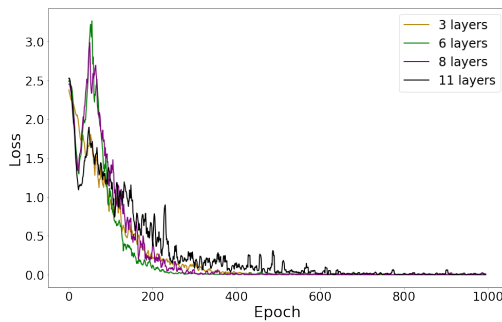


(d) 評価データの分類成功率の推移

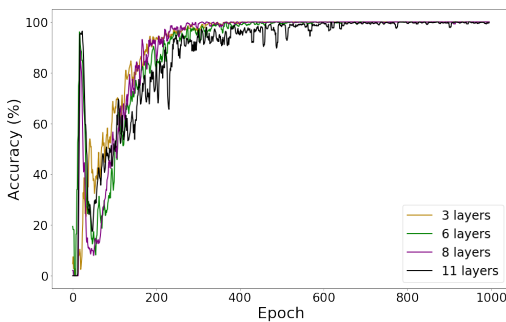
図 A.9: モンゴル語クラスの分析結果



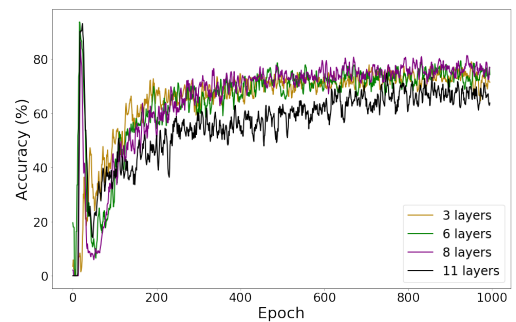
(a) Reaction 値の推移



(b) Loss 値の推移

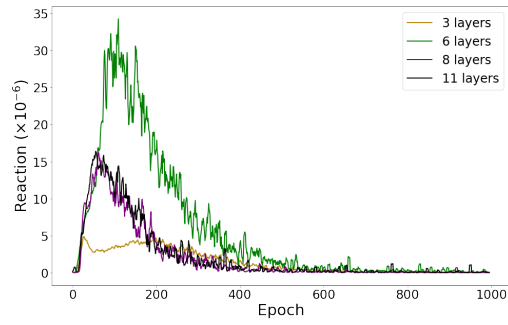


(c) 学習データの分類成功率の推移

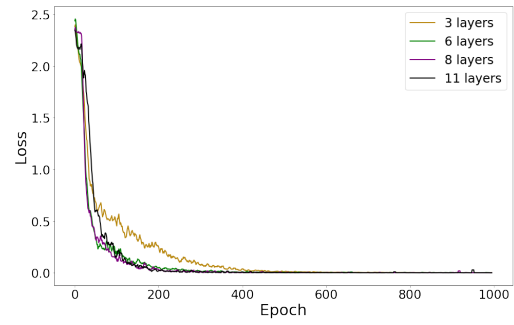


(d) 評価データの分類成功率の推移

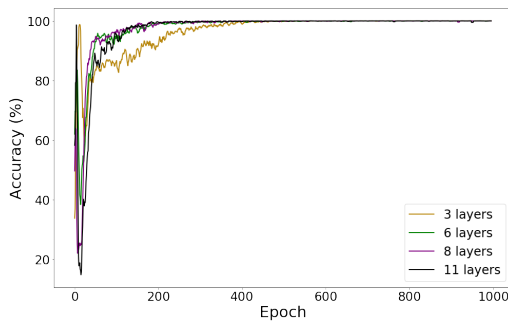
図 A.10: ロシア語クラスの分析結果



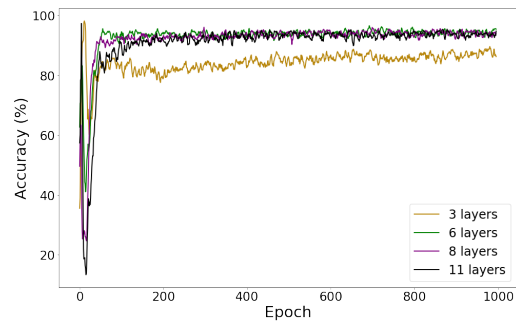
(a) Reaction 値の推移



(b) Loss 値の推移

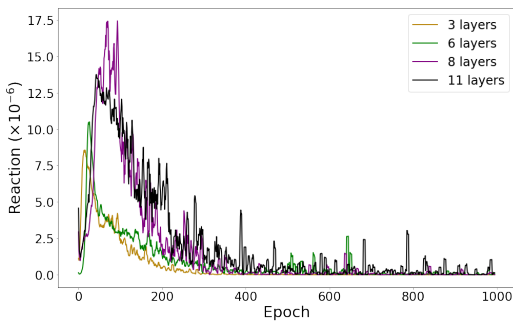


(c) 学習データの分類成功率の推移

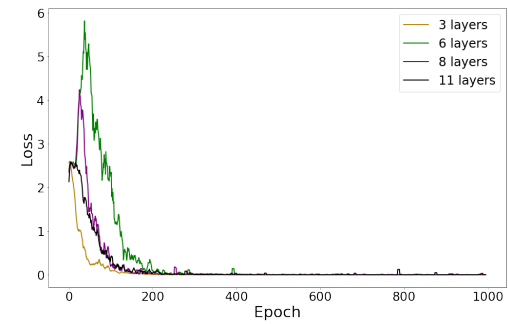


(d) 評価データの分類成功率の推移

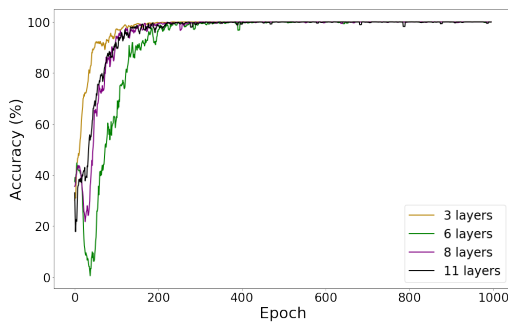
図 A.11: タイ語クラスの分析結果



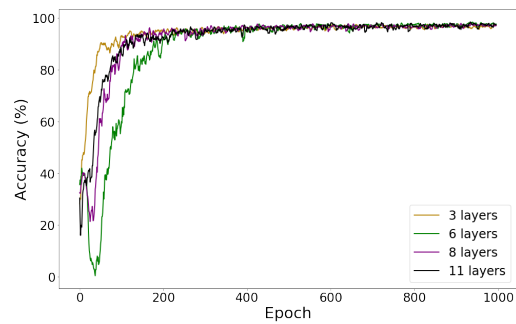
(a) Reaction 値の推移



(b) Loss 値の推移



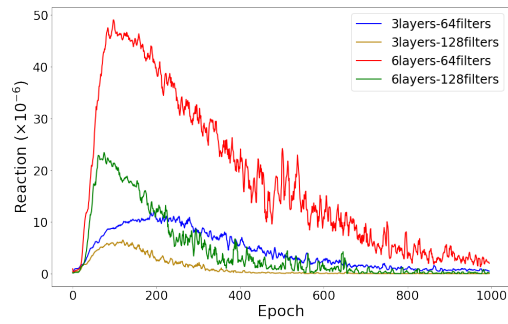
(c) 学習データの分類成功率の推移



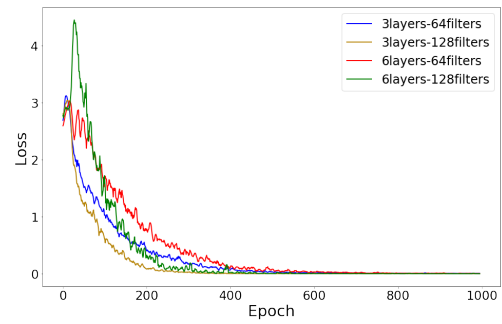
(d) 評価データの分類成功率の推移

図 A.12: チベット語クラスの分析結果

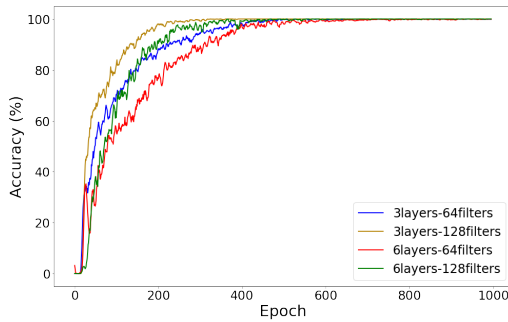
A.2 畳み込み層のフィルター枚数の変更による分析結果



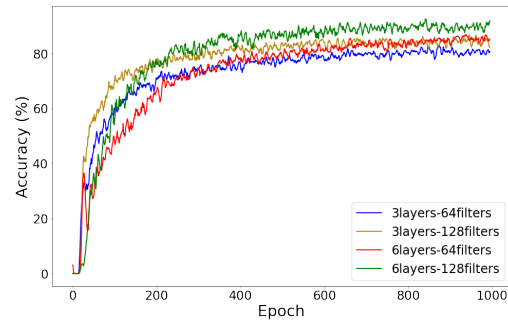
(a) Reaction 値の推移



(b) Loss 値の推移



(c) 学習データの分類成功率の推移



(d) 評価データの分類成功率の推移

図 A.13: アラビア語クラスの分析結果

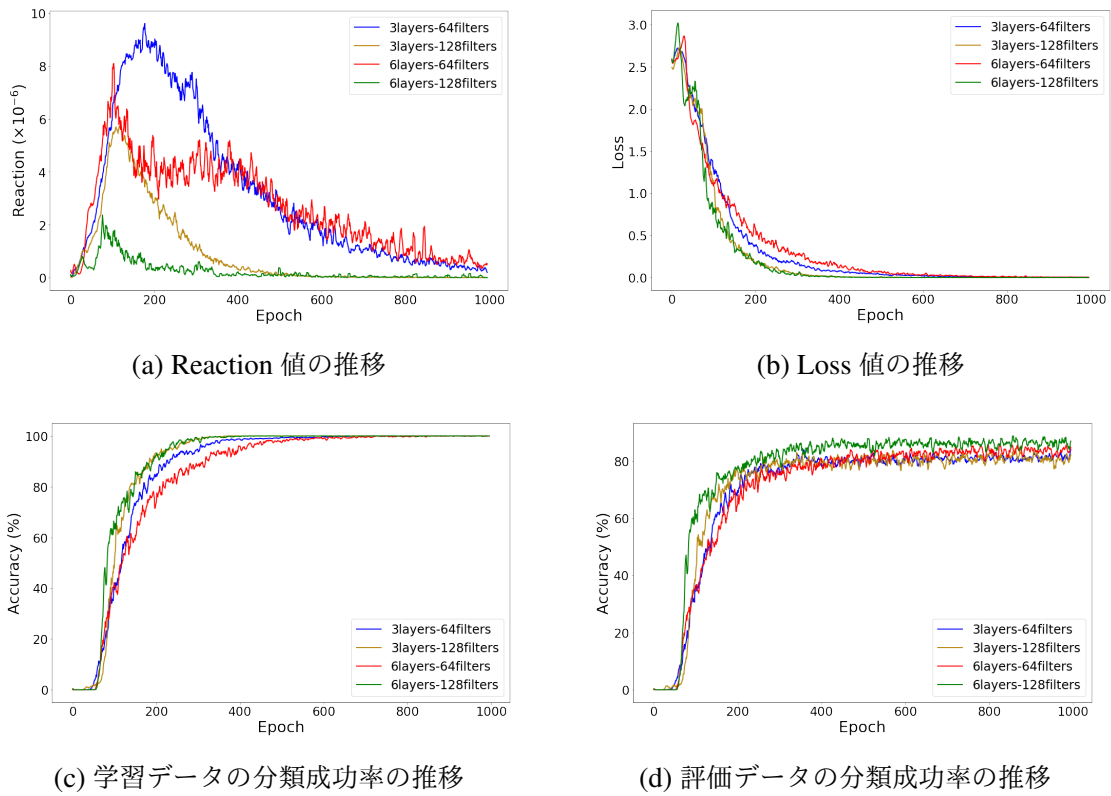


図 A.14: カンボジア語クラスの分析結果

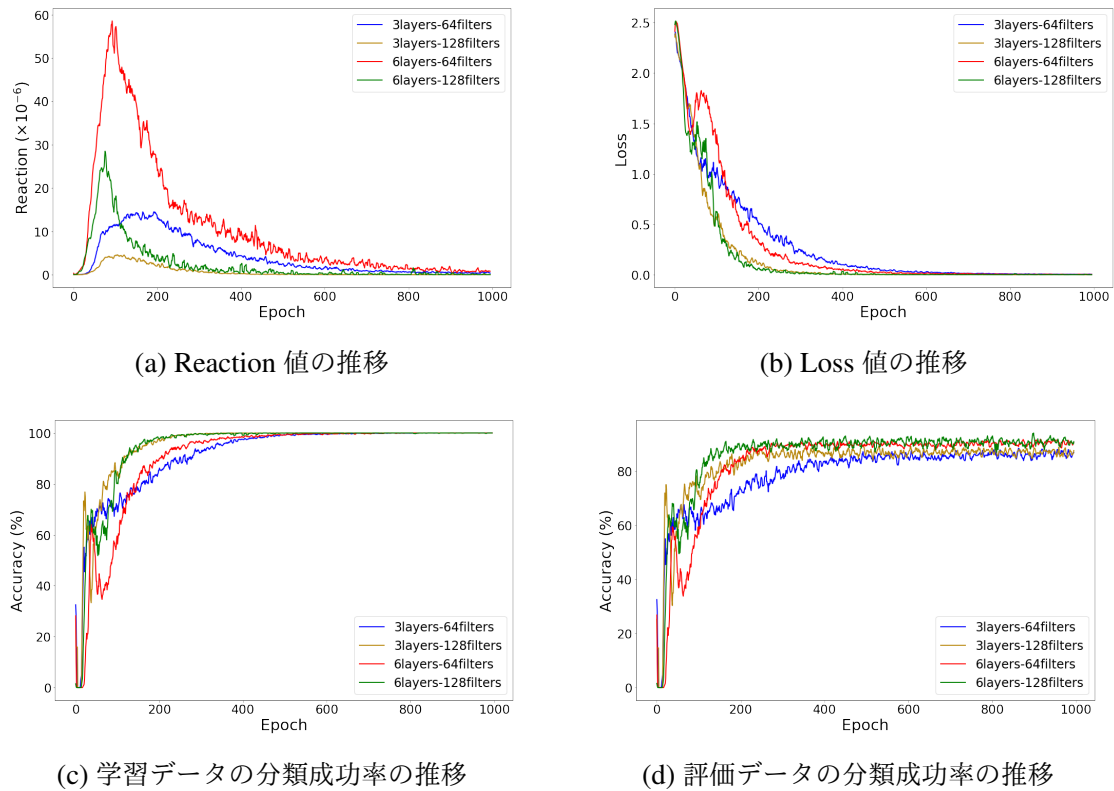


図 A.15: ヘブライ語クラスの分析結果

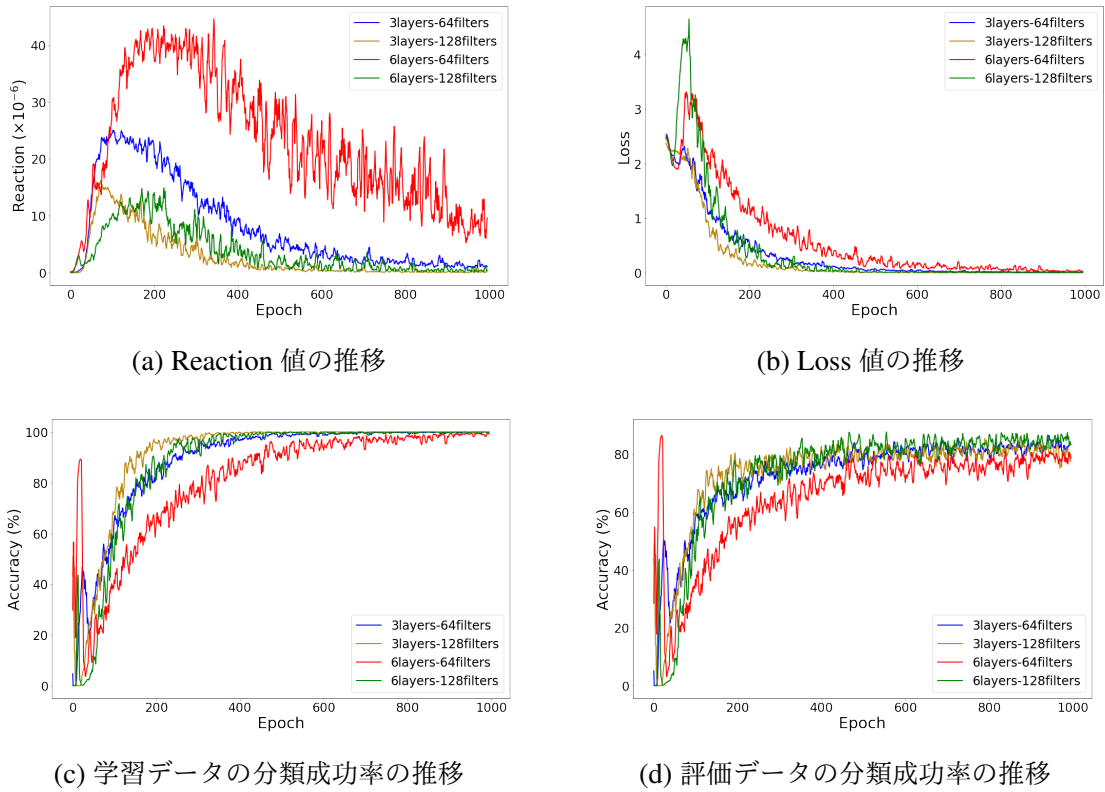


図 A.16: 日本語クラスの分析結果

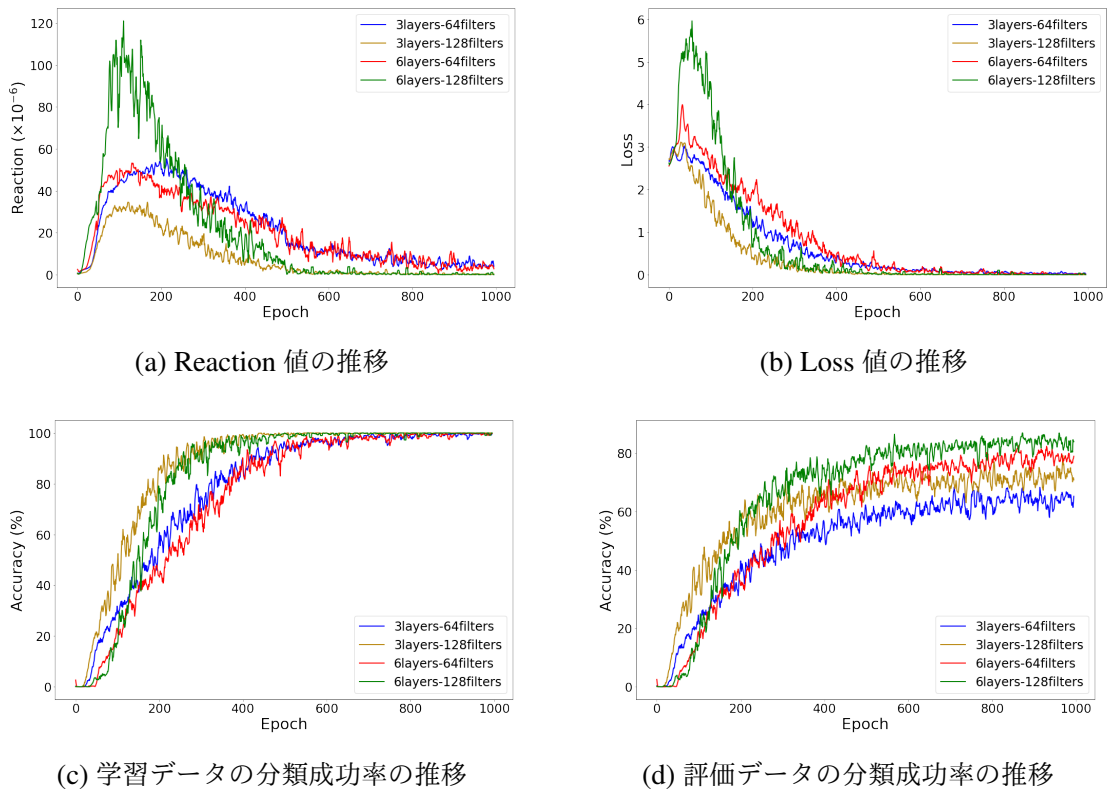


図 A.17: カンナダ語クラスの分析結果

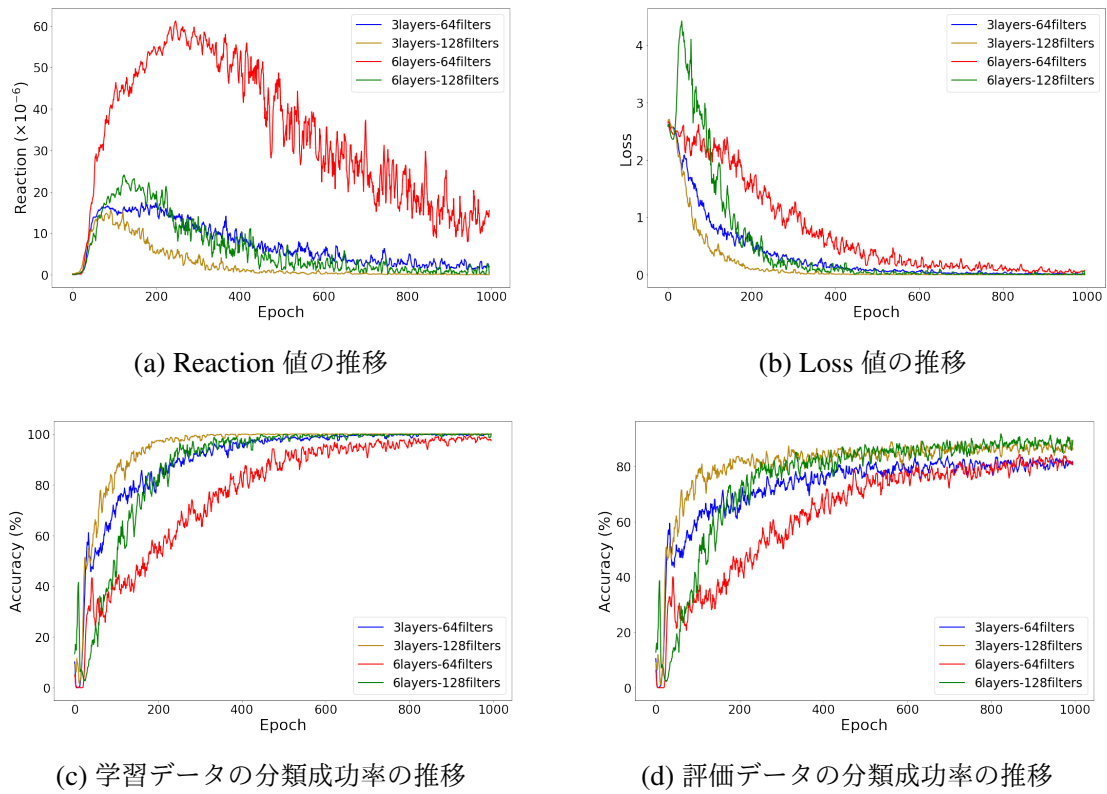


図 A.18: 韓国語クラスの分析結果

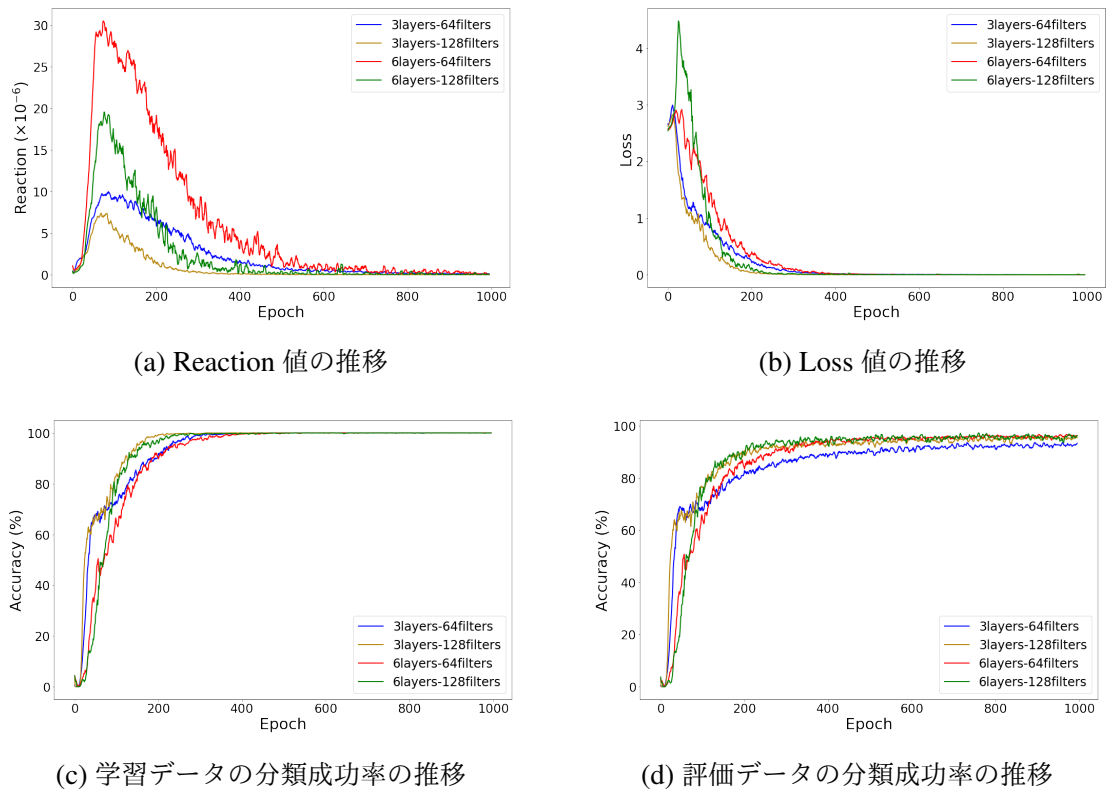
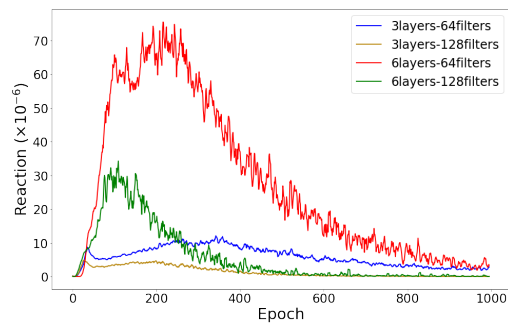
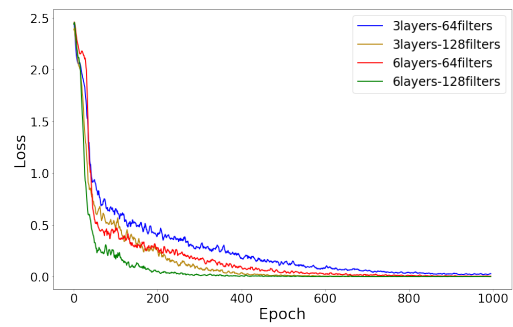


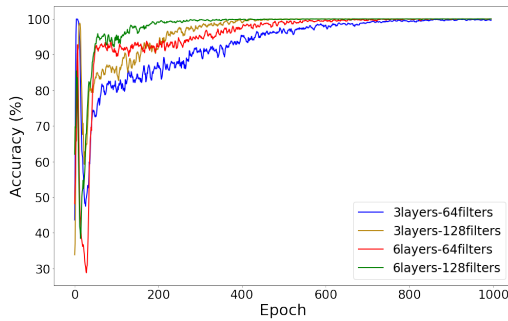
図 A.19: モンゴル語クラスの分析結果



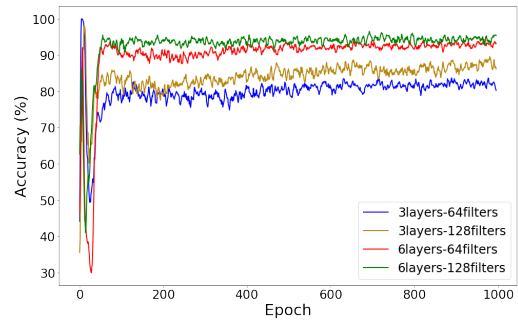
(a) Reaction 値の推移



(b) Loss 値の推移



(c) 学習データの分類成功率の推移



(d) 評価データの分類成功率の推移

図 A.20: タイ語クラスの分析結果

付録 B

検証データを用いた Reaction 値の性能評価

B.1 実験概要

4.3.2 節の分析実験では，学習データを用いた Reaction 値の性能評価を行ったが，Loss 値と同様に検証データにおいても，Reaction 値をそれぞれの CNN モデルで 1 エポックごと算出し，学習収束判定の性能を比較する．

B.2 結果と考察

まず，6 つの CNN モデルによる全 13 言語の検証データから算出した，Reaction 値の平均値の推移グラフをそれぞれ図 B.1 に示す．学習データから算出した推移グラフと異なり，Reaction 値がピークになったエポック以降の値の減少が極めて緩やかである．この傾向は，学習が活性化されてからはネットワークの学習が横ばい状態となり，勾配の変化が一定の値に収束することで，未学習の画像に対する Reaction 値も横ばいになるからだと考えられる．

次に，5.2.1 節の実験結果と，検証データにおける Reaction 値を用いて収束判定されたエポックにおける，13 言語の分類成功率を比較した結果を図 B.2 に示す．Model III では，Loss 値を用いた収束判定で得られたエポック，学習データから算出された Reaction 値，双方を用いた収束判定で得られたエポックの最大分類成功率を下回ったが，他のモデルでは，学習データを用いた Reaction 値と同等かそれ以上の収束判定の性能の高さを確認できた．

また，5.2.2 節の実験結果と，検証データにおける Reaction 値を用いて抽出されたエポックの分類成功率を比較した結果を，表 B.1 に示す．Model II，Model V，Model VI に

において、Loss 値を用いた分類成功率を上回ったが、学習データから得られた Reaction 値と比較すると、すべての CNN モデルにおいて、Loss 値を用いた分類成功率を下回った。

これらの結果より、Reaction 値は学習データ、検証データどちらを用いても、Loss 値を用いた場合より、学習過程の分析に有効であることがわかった。

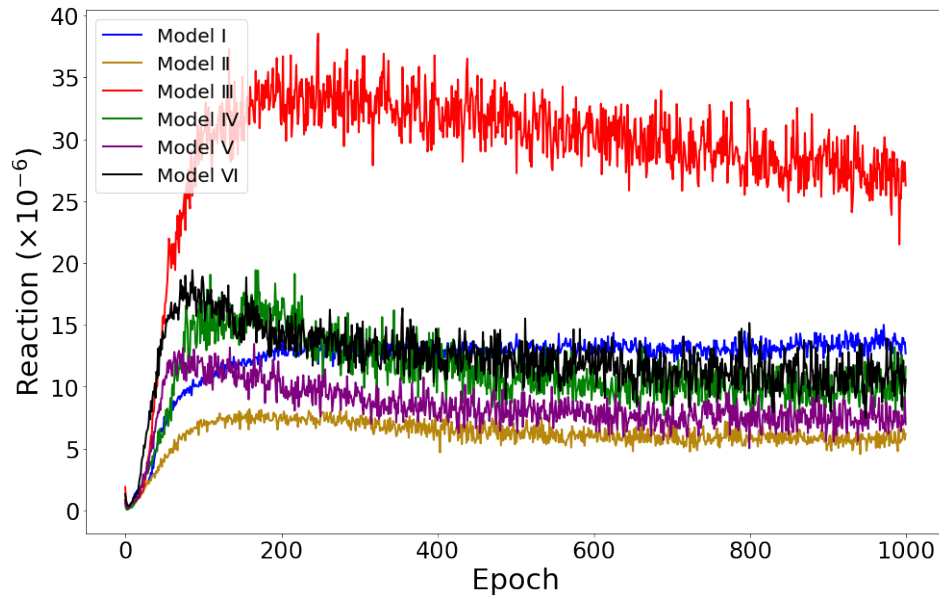
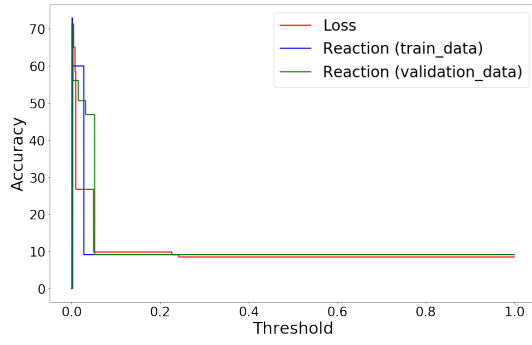


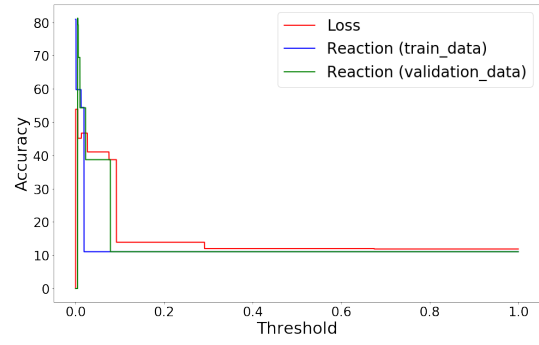
図 B.1: 6 つの CNN モデルにおける Reaction 値の推移 (検証データ)

表 B.1: 最小 Loss 値, 最小 Reaction 値をもつエポックにおける分類成功率

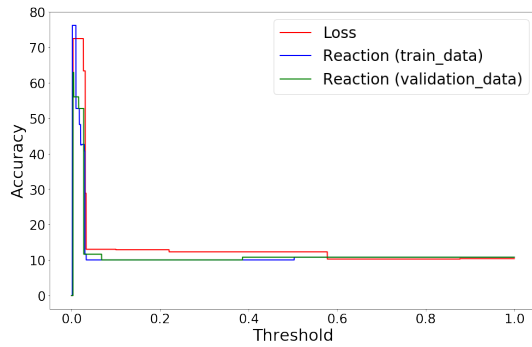
Model	Loss		Reaction (学習)		Reaction (検証)	
	Epoch	Accuracy	Epoch	Accuracy	Epoch	Accuracy
I	517	77.83	980	78.98	993	76.22
II	321	79.21	948	82.37	941	81.60
III	931	85.94	931	85.94	991	83.05
IV	919	88.85	987	89.17	907	88.17
V	154	81.72	991	89.43	799	87.95
VI	189	82.78	748	88.92	948	88.06



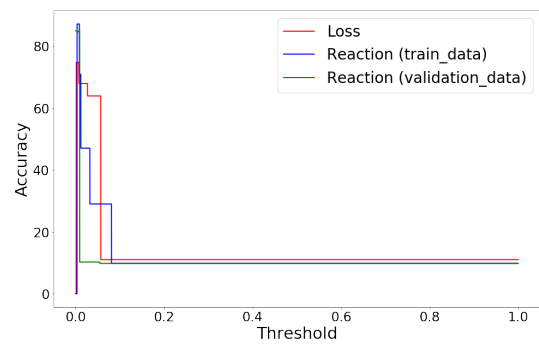
(a) Model I



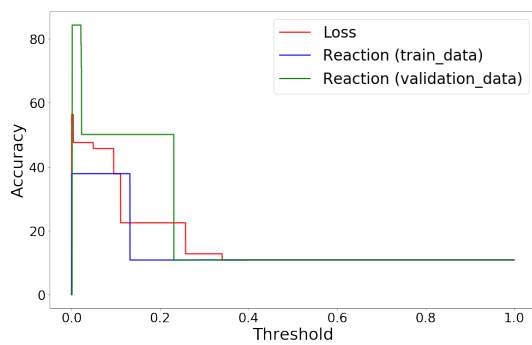
(b) Model II



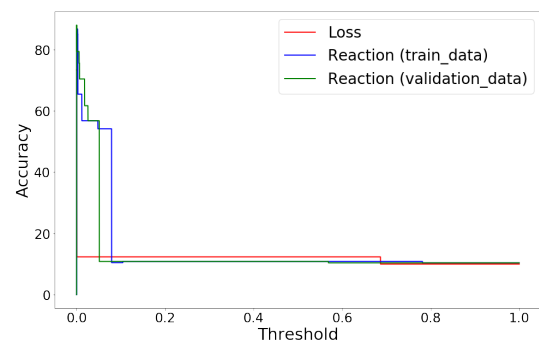
(c) Model III



(d) Model IV



(e) Model V



(f) Model VI

図 B.2: 閾値の変動による各 Model の分類成功率

付録 C

Grad-CAM を用いた CNN の判断根拠の可視化

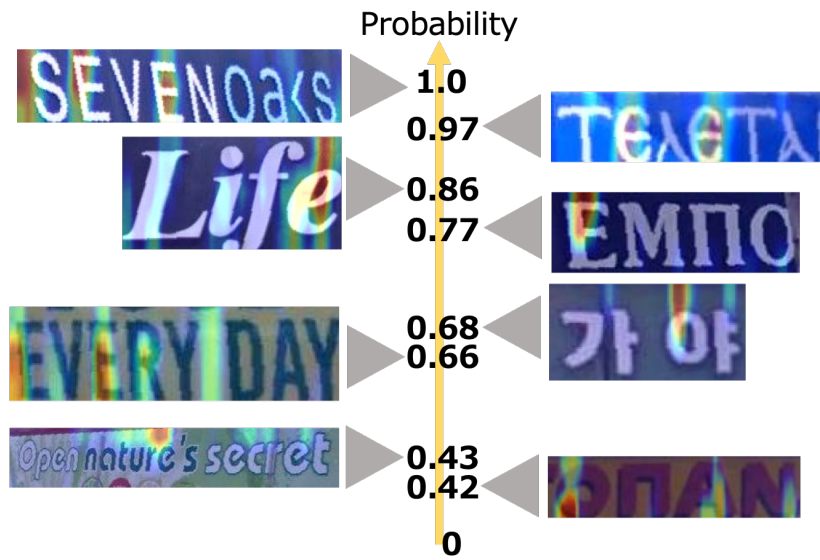
C.1 実験概要

本研究で用いた Grad-CAM を未学習の評価データに適用することで、分類の判断根拠を可視化する。使用する CNN モデルは、4.2 節の図 4.2 (c), (d) の二種を用いた。

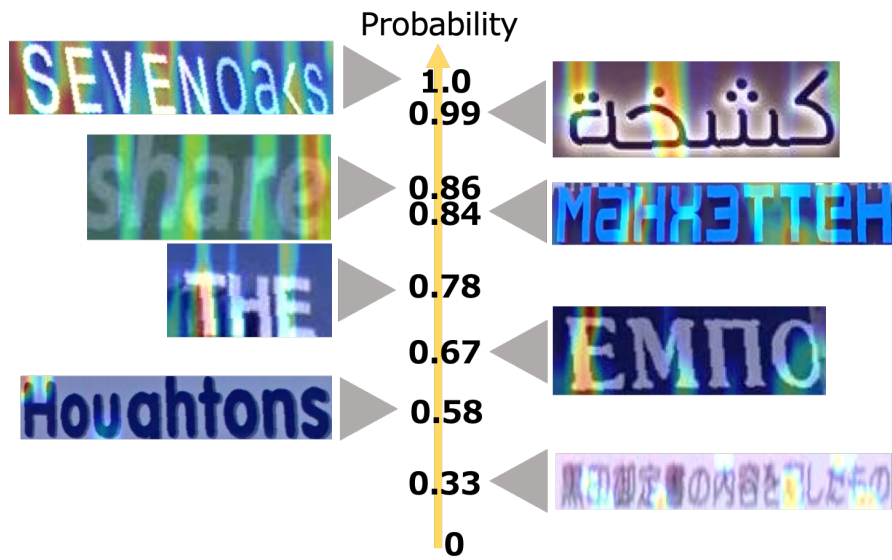
C.2 結果と考察

図 C.1 に英語クラスの評価データと、英語以外のクラスの評価データの Grad-CAM による可視化画像とその時の出力確率の結果を示す。図 C.1 (a), (b) それぞれにおいて、中心線の左側にある画像は、CNN が英語クラスとして正しく分類した文字画像である。一方で、中心線の右側にある画像は英語クラスとして誤って分類された、英語クラスではない文字画像である。また、中心線上にある確率値は、それぞれの分類画像において、英語クラスに対応する softmax 関数の出力であり、CNN がどれほど自信をもって言語を分類したかを表す指標と見なすことができる。

中心線の右側にある画像では、複数の言語で一般的に混在して使われている“E”や“T”などのアルファベット文字が原因で、英語として誤分類していることがわかる。また他の理由として、アルファベット文字の“O”や“S”などのような丸みを帯びた形にも CNN が着目していることがわかる。ゆえに、CNN は英語ではない言語に対して英字に似た特徴に誤って着目してしまうため、英語クラスに対する確率値が大きくなり、結果として英語として誤分類されたと考えられる。よって、言語判定の失敗の理由の一つとして、他言語の文字の特徴に対して CNN が誤った着目をしてしまうことがこの可視化結果から示唆される。



(a) Model IV



(b) Model V

図 C.1: Grad-CAM による可視化結果と出力確率

付録 D

研究で用いたデータの参照場所

研究に用いたすべてのプログラムコードはヒューマンインタフェース研究室サーバー内の

- /net/xserve0/users/tomioka/Master

のディレクトリ下に存在する。本ディレクトリの構成は以下のとおりである。

```
Master/ # 研究用ディレクトリ
|
|-- Codes/ # 本研究で作成したコード保存用ディレクトリ
|
|-- Datas/ # 画像, 実験データの保存用ディレクトリ
|
|-- Results/ # 本論文で示した実験データ用ディレクトリ
```

詳しくは本ディレクトリ下の `readme.md` に示す。


付録 E

発表資料

修士論文発表で用いた発表資料を以下に掲載する。

**CNNを用いた言語判定における
学習過程の分析**

三重大学大学院 工学研究科 情報工学専攻
 ヒューマンインターフェース研究室
 418M518 富岡永伍

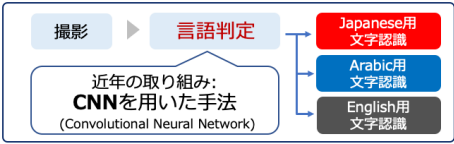


研究背景

社会・経済のグローバル化

- さまざまな言語の文字認識技術が必要である。
- 文字認識の前に画像中の文字の言語だけを判定する。
- 精度の向上, 学習コストの削減などが期待できる。

言語判定処理



撮影 ▶ **言語判定**

近年の取り組み:
CNNを用いた手法
 (Convolutional Neural Network)

Japanese用
文字認識
 Arabic用
文字認識
 English用
文字認識


研究背景

Convolutional Neural Networks (CNN)

- さまざまな画像問題において高い性能を発揮する。
- **弱み:** 学習の過程が不透明である。
誤分類をした時の原因考察が困難。

ブラックボックス化された分類器として使われる。

- CNNをより良く理解するための取り組み
 - ▶ 分類の判断根拠を可視化する手法
 - ▶ 可視化によるネットワークの性能比較
 - ▶ 学習のパラメータの最適化手法



[1] Karen Simonyan et al. "Deep inside convolutional net-works: Visualising image classification models and saliency maps" ICLR, 2014.

関連研究と本研究の目的

CNNの学習の最適化やネットワーク同士の比較手法

- CNNの特徴マップの相関関係を比較する手法 [2]
- パラメータの違いによる損失関数の曲率の比較手法 [3]
- 学習中の畳み込み層の飽和度を算出する手法 [4]

本研究の目的とアプローチ

CNNの学習過程の理解
(損失関数 (Loss) を用いた場合より有効な分析を目指す.)

新たにCNNの学習過程の分析手法を提案する。
(本研究では言語判定問題における学習過程に着目する.)

[2] M. Raghu et al. "SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability" Advances in NIPS, 30 2017.
 [3] H. Li et al. "Visualizing the loss landscape of neural nets" Advances in NIPS, 31, 2018.
 [4] J. Shenk et al. "Spectral Analysis of Latent Representations" arXiv:1907.08589, 2019.

本研究の成果

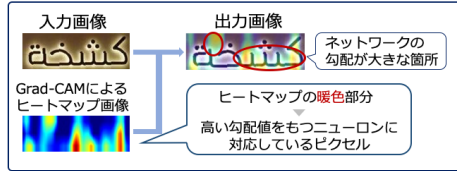
- ハイパーパラメータ変更による学習過程の調査
 - Loss値に代わる数値指標を定義することで、クラスごとの学習進行状況を比較した。
- **学習エポックの収束判定に関する実験**
 - Loss値を用いた収束判定と比較し優位性を確認した。
- 査読有 国際会議1件, 国内会議1件で発表
 - E.Tomioka et al. Proceedings of ICITR2019, (December, 2019)
 - 富岡 他, MIRU2019, (7/29~8/1, 2019)

本発表の内容

関連手法

CNNの畳み込み層の勾配値の計算手法

- **Grad-CAM** (Gradient-weighted Class Activation Mapping) [5]
 - ネットワークの勾配 (各ピクセルにおける分類時の注目度を示す指標) を求めるための主要な技術である。



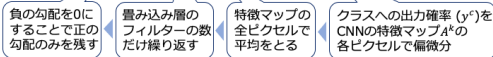
[5] Zhou Bolei, et.al. "Learning deep features for discriminative localization", IEEE, 2016

関連手法

Grad-CAMの計算処理の流れ

- 各ピクセルに対して**勾配値** (ニューロンの活性化箇所)を算出する。

$$L^c = \max \left(0, \sum_k \frac{1}{WH} \sum_{i=1}^W \sum_{j=1}^H \frac{\partial y^c}{\partial A_{ij}^k} A_{ij}^k \right)$$



本研究ではGrad-CAMによる**勾配値**を応用することでCNNの学習進捗を調査する。

提案手法

Grad-CAMは本来、未学習の画像 (評価画像)に対して使用される。

本研究ではあえて学習画像にGrad-CAMを適用する。

学習画像における**勾配値**の平均値を算出することで学習の進み具合が明らかになる。

各ピクセルの勾配値を画像中で**平均化**することで (**Reaction値**と定義) クラスごとの学習進捗を調査する。

$$Reaction = \frac{1}{N} \sum_i \sum_j L_{ij}^c \quad (N: \text{正の勾配値をもつピクセルの数})$$

評価実験 -概要-

- **SIW-13 dataset** [6]
 - 13言語の情景内画像からテキスト領域を切り出したデータセット
 - 学習データ: 9,791枚, 評価データ: 6,500枚 (各言語500枚) (本実験では学習データの1割を**検証データ**として使用する。)



[6] Zhou Bolei, et.al. "Learning deep features for discriminative localization", IEEE, 2016

評価実験 -評価方法-

- 1000エポック学習した構造の異なる**6種類**のCNNを用いて Reaction値とLoss値の有用性をそれぞれ比較する。

実験1: 学習過程の比較

- Reaction値, Loss値 (学習データ)それぞれの推移を各CNNで比較。

実験2: 学習収束判定の性能比較 (検証データのLossと比較する。)

- 下記の条件を満たすエポック (i) を抽出し, 収束判定の汎用性を比較。

条件1 (値の差分の収束)

$$(V_{i+1} - V_i) / S_i < \text{閾値 (threshold)} \quad (0 < \text{閾値} \leq 1)$$

条件2 (値の減少)

$$\sum_{i=n-1}^{n+4} V_i > \sum_{i=n}^{n+5} V_i$$

(V: Reaction値またはLoss値, S: Reaction値またはLoss値の標準偏差)

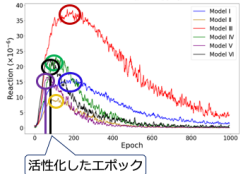
実験3: 分類成功率の比較 (検証データのLossと比較する。)

- Reaction値, Loss値で最小値をもつエポックにおいて分類成功率を比較。

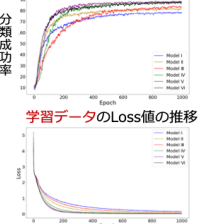
評価実験 -実験1結果-

学習過程の比較

全学習データの平均Reaction値を算出、Reaction値の推移



評価データの分類成功率の推移



- 各モデルで**学習が活性化**したエポックを分析しやすい。
- Loss値と比べてモデル同士の**学習過程の比較**が容易である。

評価実験 -評価方法-

- 1000エポック学習した構造の異なる**6種類**のCNNを用いて Reaction値とLoss値の有用性をそれぞれ比較する。

検証データを多く用いるほど、学習データ数は減る。
→ 検証データを用いなくても十分な学習過程の分析を行いたい。

実験2: 学習収束判定の性能比較 (検証データのLossと比較する。)

- 下記の条件を満たすエポック (i) を抽出し, 収束判定の性能を比較。

条件1 (値の差分の収束)

$$(V_{i+1} - V_i) / S_i < \text{閾値 (threshold)} \quad (0 < \text{閾値} \leq 1)$$

条件2 (値の減少)

$$\sum_{i=n-1}^{n+4} V_i > \sum_{i=n}^{n+5} V_i$$

(V: Reaction値またはLoss値, S: Reaction値またはLoss値の標準偏差)

実験3: 分類成功率の比較 (検証データのLossと比較する。)

- Reaction値, Loss値で最小値をもつエポックにおいて分類成功率を比較。

評価実験 -実験2結果-

学習収束判定の性能比較 学習データのReaction 検証データのLoss

- 条件1: 値の差分の収束
- 条件2: 値の減少

条件を満たすエポックの分類成功率を比較する。

収束と判定されたエポックにおける評価データでの分類成功率

青: Reaction値
赤: Loss値

評価実験 -実験2結果-

学習収束判定の性能比較 学習データのReaction 検証データのLoss

- 条件1: 値の差分の収束
- 条件2: 値の減少

条件を満たすエポックの分類成功率を比較する。

Reaction値は学習データから得られる値でも学習が十分に進んだエポックを抽出することができる。

収束と判定されたエポックにおける評価データでの分類成功率

青: Reaction値 拡大
赤: Loss値

評価実験 -評価方法-

- 1000エポック学習した構造の異なる6種類のCNNを用いてReaction値とLoss値の有用性をそれぞれ比較する。

実験1: 学習過程の比較

> Reaction値, Loss値 (学習データ)それぞれの推移を各CNNで比較。

実験2: 学習収束判定の性能比較 (検証データのLossと比較する。)

> 下記の条件を満たすエポック (i) を抽出し, 収束判定の汎用性を比較。

条件1 (値の差分の収束)

$$\frac{(V_{i+1} - V_i)}{S_i} < \text{閾値 (threshold)}$$

(0 < 閾値 ≤ 1)

条件2 (値の減少)

$$\sum_{i=n-1}^{n+4} V_i > \sum_{i=n}^{n+5} V_i$$

(V: Reaction値またはLoss値, S: Reaction値またはLoss値の標準偏差)

実験3: 分類成功率の比較 (検証データのLossと比較する。)

> Reaction値, Loss値で最小値をもつエポックにおいて分類成功率を比較。

評価実験 -実験3結果-

分類成功率の比較 ※ Reaction値: 学習データによる値を使用。 Loss値: 検証データによる値を使用。

値がピークになったエポック以降で最小値を抽出する。

最小値をもつエポックで評価データの分類成功率を比較。

評価実験 -実験3結果-

分類成功率の比較 ※ Reaction値: 学習データによる値を使用。 Loss値: 検証データによる値を使用。

値がピークになったエポック以降で最小値を抽出する。

6つ全てのCNNにおいてReaction値の方が高い分類成功率をもつエポックを抽出した。

Model	Epoch	Loss		Reaction	
		成功率	Epoch	成功率	Epoch
I	517	77.83	980	78.98	
II	321	79.21	948	82.37	
III	931	85.94	931	85.94	
IV	919	88.85	987	89.17	
V	154	81.72	991	89.43	
VI	189	82.78	748	88.92	

Reaction値は検証データを用いなくても学習過程の分析に有効である。

まとめ

- CNNの学習進行状況を分析する手法を提案した。
- Reaction値を用いた学習過程の分析の特徴
 - > 異なるネットワーク同士でも学習過程の比較が容易である。
 - > 学習が活性化したエポックを明らかにできる。
 - > Loss値より最適なエポックの抽出が可能である。
- 今後の課題
 - > 計算コストの削減 (扱うCNNの規模や学習データ数に依存)
 - > 言語判定データセット以外への適用

謝辞

本論文の執筆，また私の研究生活は多くの方々のご支援をいただきました。若林哲史教授には，本研究を進めるにあたり，適切な御助言，ご指導を幾度となくしていただきました。盛田健人助教には，着任1年目にも関わらず私の研究に深く携わってくださり，多くのアイデアを提供していただきました。白井伸宙助教には，日頃からのディスカッションによって多くのことを学ばせていただき，学会のための論文添削にも深く携わっていただきました。また，三宅康二名誉教授には，お忙しい中ディスカッションに参加して下さり，積極的にアドバイスしていただきました。そして，埼玉工業大学の大山航教授には，三重大学を離れられてからも私の研究に対するアドバイスを熱心にしていただきました。私の研究に携わっていただいた先生方に心より感謝申し上げます。また，諸連絡の掲示や備品の貸し出しなど，日頃様々なお世話をしていただき研究を快適にしやすい環境を作ってくれた吉永みゆき事務員に深く感謝いたします。そして，本研究や研究室生活への多くのアドバイスを与えて下さった研究室の先輩，後輩の皆様，お互いの研究について議論し合い，技術の共有を行い，共に切磋琢磨した同期の皆様に感謝します。最後になりましたが，私の学生生活を最後まで支えていただいた家族に今一度感謝の意を表して，本論文の結びといたします。

参考文献

- [1] TN Tan: “Rotation invariant texture features and their use in automatic script identification” IEEE PAMI, Vol.20, pp.751-756, (1998)
- [2] Anguelos Nicolaou, Andrew Bagdanov, Lluís Gomez-Bigorda, and Dimosthenis Karatzas, “Visual Script and Language Identification” 2016 12th IAPR Workshop on Document Analysis Systems (DAS), on pp.393-398, (2016)
- [3] Ankan Kumar Bhunia, Aishik Konwer, Abir Bhowmick, Ayan Kumar Bhunia, and Partha P. Roy: “Script Identification in Natural Scene Image and Video Frame using Attention based Convolutional-LSTM Network” Pattern Recognition, Vol.85, pp.172-184, (2019)
- [4] Yash Patel, Michal Busta, and Jiri Matas: “E2E-MLT-an unconstrained End-to-end method for multi-language scene text” arXiv preprint arXiv:1801.09919, (2018)
- [5] LeCun Yann, Leon Bottou, Yoshua Bengio, and Patrick Haffner: “Gradient-based learning applied to document recognition” Proceedings of the IEEE, Vol.86, Issue.11, pp.2278-2324, (1998)
- [6] Krizhevsky Alex, Sutskever Ilya, and Hinton Geoffrey E: “Imagenet classification with deep convolutional neural networks” In Advances in neural information processing systems, pp. 1097-1105, (2012)
- [7] Matthew D.Zeiler, Rob Fergus: “Visualizing and understanding convolutional networks” In Computer Vision-ECCV 2014, pp. 818-833, (2014)
- [8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell: “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition” ICML’14 Proceedings of the 31st International Conference on International Conference on Machine Learning, Vol.32, pp.647-655, (2014)
- [9] Jiuxiang Gao, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Li Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen: “Recent Advances in Convolutional Neural Networks” Pattern Recognition,

- Vol.77, pp.354-377, (2018)
- [10] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry: “How Does Batch Normalization Help Optimization?”, *Advances in Neural Information Processing Systems* 31, pp.2488-2498, (2018)
- [11] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang: “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima” In *The International Conference on Learning Representations*, (2017)
- [12] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman: “Deep inside convolutional net-works: Visualising image classification models and saliency maps” *International Conference on Learning Representations Workshop*, (2014)
- [13] Ramprasaath R Selvaraju, Abhishek Das: Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra: “Grad-cam: Why did you say that?” *arXiv preprint arXiv:1611.07450*, (2016)
- [14] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viegas, and Martin Wattenberg: “SmoothGrad: removing noise by adding noise” *arXiv preprint arXiv:1706.03825*, (2017)
- [15] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba: “Interpretable Basis Decomposition for Visual Explanation” *European Conference on Computer Vision*, pp.122-138, (2018)
- [16] Noboru Murata, Shuji Yoshizawa, and Shun-ichi Amari: “Network information criterion-determining the number of hidden units for an artificial neural network model” *IEEE Trans. Neural Networks*, Vol.5, pp.865-872, (1994)
- [17] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein: “SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability” *Advances in Neural Information Processing Systems* 30, pp.6078-6087, Curran Associates, (2017)
- [18] D.R.Hardoon, S.Szedmak, and J.Shawe-Taylor: “Canonical correlation analysis: An overview with application to learning methods” *Neural Computation*, Vol.16, Issue.12, pp.2639-2664, (2004)
- [19] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein: “Visualizing the loss landscape of neural nets” *Advances in Neural Information Processing Systems* 31, pp.6389-6399, Curran Associates, (2018)
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and JianSun: “Deep Residual Learning for Image Recognition” In *Proceedings of CVPR*, pp.770-778, (2016)

- [21] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten: “Densely connected convolutional networks” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4700-4708, (2017)
- [22] Justin Shenk, Mats L. Richter, Anders Arpteg, and Mikael Huss: “Spectral Analysis of Latent Representations” arXiv preprint arXiv:1907.08589, (2019)
- [23] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams: “Learning representations by back-propagating errors” Nature 323, pp.533-536, (1986)
- [24] Yann Le Cun, L. Bottou, and Y. Bengio: “Reading checks with multilayer graph transformer networks” 1997 IEEE International Conference on Computer Vision and Pattern Recognition, pp.3431-3440, (1997)
- [25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation” In International Conference on Medical image computing and computer-assisted intervention, pp. 234-241. Springer, (2015)
- [26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp.3431-3440, (2015)
- [27] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial nets” In Advances in neural information processing systems, pp.2672-2680, (2014)
- [28] Ramprasaath R.Selvaraju, Michael Congswell Devi Parikh, Adhishek Das Dhruv Batra, and Ramakrishna Vedantam: “Grad-CAM : Visual Explanations from Deep Networks via Gradient-based Localization” The IEEE International Conference on Computer Vision (ICCV), pp.618-626, (2017)
- [29] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba: “Learning Deep Features for Discriminative Localization” Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. IEEE, (2016)
- [30] Min Lin, Qiang Chen, and Shuicheng Yan: “Network In Network” International Conference on Learning Representations (ICLR), (2014)
- [31] Baoguang Shi, Xiang Bai, and Cong Yao: “Script identification in the wild via discriminative convolutional neural networks” Pattern Recognition, Vol.52, pp.448-458, (2016)
- [32] Google Street View. <<http://maps.google.com> >
- [33] Dimosthenis Karatzas, S. Robles Mestre, Juan mas romeu, Farshad Nourbakhsh, and P. Pratim Roy: “ICDAR 2011 robust reading competition-challenge 1: reading text in born-digital images (web and email)” in Document Analysis and Recognition (ICDAR), 2011 International Conference on. IEEE, pp. 1485-1490, (2011)

-
- [34] Kai Wang, and Serge Belongie: “Word Spotting in the Wild” European Conference on Computer Vision (ECCV) 2010, Computer Vision - ECCV 2010, pp 591-604, (2010)
- [35] Anand Mishra, Karteek Alahari, and C. V. Jawahar: “Scene Text Recognition using Higher Order Language Priors” In Proceedings British Machine Vision Conference (BMVC) 2012. pp.1-11, (2012)
- [36] Sergey Ioffe, Christian Szegedy: “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift” The 32nd International Conference on International Conference on Machine Learning (ICML), Vol.37, pp.448-456, (2015)